Created by: Mohamed Faris Bin Mohd Yazid
Date created: 9 June 2025
Test: 200-301 CCNA v1.1

This is a summary I have created when studying for the CCNA exam. The notes are mostly from JITL CCNA youtube course and are meant to be used only after watching them. I hope that this will benefit anyone who is studying for the CCNA. Do feel free to drop me an email at farisyazid1702@gmail.com if you have comments or suggestions to give me. Thanks

Jeremy's IT Lab - CCNA course:
https://www.youtube.com/playlist?list=PLxbwE86jKRgMpuZuLBivzlM8s2Dk5lXBQ

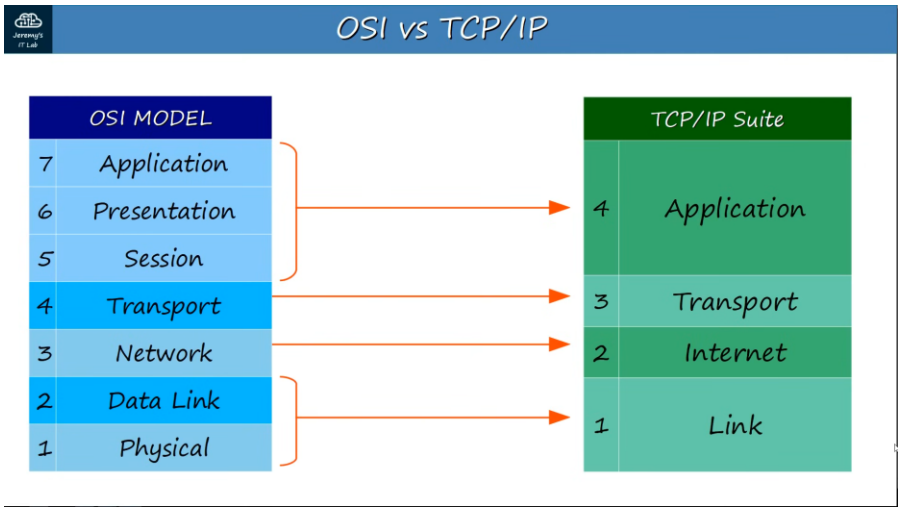
Topics not included
- Routing Fundamentals: not much to memorise, just understanding
- Subnetting
- WLC config (only some notes, not actual config)

OSI

- 7 layers
    - Application
        - Identifying communication partners
        - Synchronising communication
    - Presentation
        - Translate btw different application layer format
    - Session
        - Control dialogues (sessions) btw communicating hosts
    - Transport
        - Segments and reassembles data
        - Provide host-to-host communication
    - Network
        - Path selection
        - Logical addressing (IP address)
        - Provide connectivity btw end hosts on different networks
    - Data Link
        - Provide node-to-node connectivity
        - Detect and correct Layer 1 errors
        - Layer 2 addressing (MAC address)
    - Physical
        - Physical characteristics of the medium used

Protocol Data Units (PDU)

- Layer 7 -5: Data
- Layer 4: Segment
- Layer 3: Packet
- Layer 2: Frame
- Layer 1: Bit



Intro to CLI

Connecting to device

- Roll-over cable
- Putty Settings (default settings)
    - Serial
    - Speed (baud) : 9600
    - Data bits: 8
    - Stop bits: 1
    - Parity: None
    - Flow Control: None

User Modes

- User EXEC mode
- Privileged EXEC mode
- Global Config mode

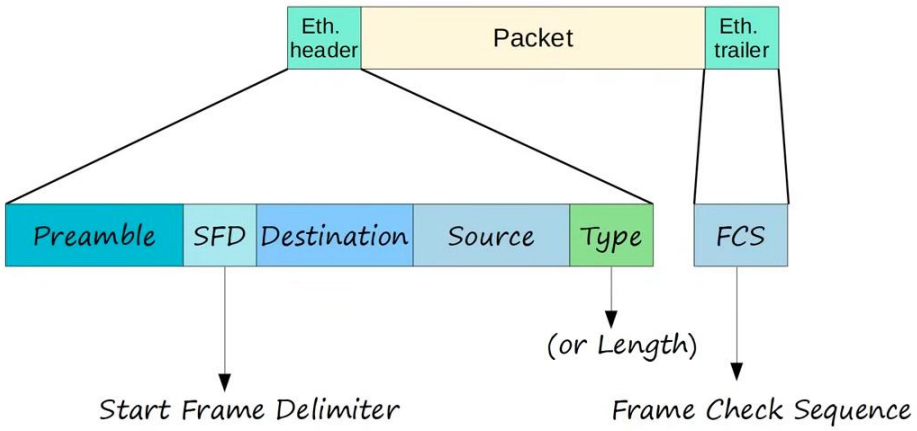
Ethernet

Ethernet Standard

Speed	Common Name	IEEE Standard	Informal Name	Maximum Length
10 Mbps	Ethernet	802.3i	10BASE-T	100 m
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100 m
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100 m
10 Gbps	10 Gig Ethernet	802.3an	10GBASE-T	100 m

Informal Name	IEEE Standard	Speed	Cable Type	Maximum Length
1000BASE-LX	802.3z	1 Gbps	Multimode or Single-Mode	550 m (MM) 5 km (SM)
10GBASE-SR	802.3ae	10 Gbps	Multimode	400 m
10GBASE-LR	802.3ae	10 Gbps	Single-Mode	10 km
10GBASE-ER	802.3ae	10 Gbps	Single-Mode	30 km

Ethernet Frame



- Preamble
  - 10101010
- SFD
  - 10101011
- Type
  - >1536: type
  - <= 1500: size (Max size = 1500)
  - IPv4: 0x0800
  - IPv6: 0x86DD
  - ARP: 0x0806
- Minimum size
  - Without Preamble & SFD: 64B
  - Min payload (w/o header): 46B
  - If smaller than 46B, padding (0s) added

MAC Address

- 'Burned In Address' (BIA)
- First 3 bytes: OUI (Organizationally Unique Identifier)

Ethernet LAN Switching

- ARP (Address Resolution Protocol)
  - ARP Request: Broadcast
  - ARP Reply: Unicast
- Ping
  - ICMP Echo Request: Unicast
  - ICMP Echo Reply: Unicast

- ARP
  - Used to find the MAC address of a known IP address
- Unknown unicast flooding

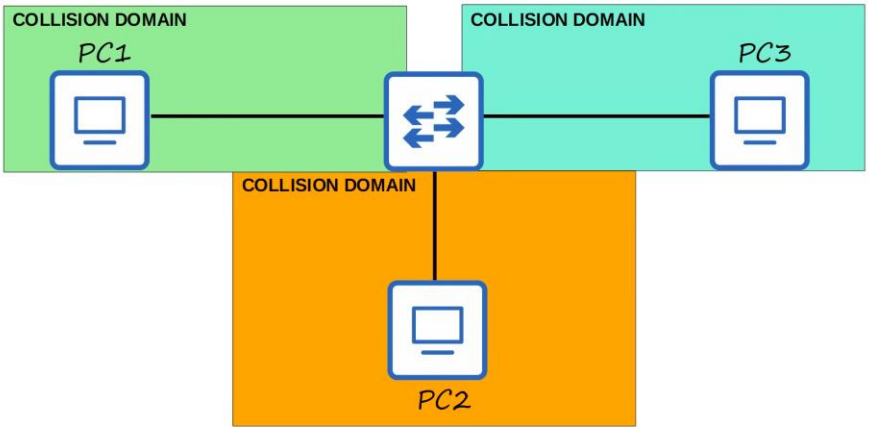
- Used when the destination MAC address is unknown

Switch Interfaces

CSMA/CD

- Carrier Sense Multiple Access/ Collision Domain
- Device listen to collision domain before sending
- Send jamming signal if collision occur
- Each device wait a random time and resend
- Process repeats

Switch Collision Domain



Speed/Duplex Auto-Negotiation

- Interfaces default to auto speed and duplex
- Interface advertise capabilities to neighbour, try to match settings that both can
- What if autonegotiation is disabled?
    - Speed
        - The switch will try to sense the speed that the other device is operating at
        - If cannot, it will use the slowest supported speed
    - Duplex
        - If speed is 10 or 100 Mbps, half duplex
        - If 1000 or more, full duplex
- Duplex mismatch, collision will occur

Errors

- Runts: too small
- Giants: too big
- CRC: fail check
- Frame: Incorrect format
- Input error: total count of 4 above
- Output errors: failed to send

IPv4 Addressing

IP header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

- Version (4 bit)
- IHL (Internet Header Length) (4 bit)
    - 4-byte increment
    - Min: 5 = 20B
    - Max: 15 = 60B
- DSCP (Differentiated Services Code Point) (6 bit)
    - QoS
- ECN (Explicit Congestion Notification) (2 bit)
    - Notify of network congestion w/o dropping packets
- Total Length (16 bit)
    - Min: 20 (IPv4 header only)
    - Max: 65,535
- Identification Field (16 bit)
    - Identify which packet fragments belong to
- Flags Field (3 bit)
    - Bit 0: reserved, always 0
    - Bit 1: DF bit
    - Bit 2: MF bit
- Fragment Offset (13 bit)

    - Identify position of fragment in original packet
- TTL (Time To Live) (8 bit)
    - Prevent infinite loops
    - Dropped if 0
    - Used as hop count
    - Recommended default = 64
- Protocol (8 bit)
- Header Checksum (16 bit)
- Source/Destination IP Address (32 bit)
- Options (0-320 bit)

VLAN

- 5 default VLAN exist
    - 1
    - 1002-1005
- Access port
    - An access port that belongs to 1 VLAN
- Trunk ports
    - An access port that belongs to more than 1 VLAN
    - Frames will be tagged

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

- TPID (Tag Protocol Identifier)
    - Always 0x8100
    - Indicated 802.1q tagged
- TCI (Tag Control Information)
- PCP (Priority Code Point)
    - For class of service, prioritize important traffic in congestion
- DEI (Drop Eligibility Indicator)
- VID (VLAN ID)
- VLAN range

    - Normal: 1 - 1005
    - Extended: 1006 - 4094
    - Reserved: 0, 4095
    - ISL: 1 - 4094
- Native VLAN
    - Only supported by 802.1q
    - Frames from native VLAN won't be tagged
    - The switches that are connected must have same native VLAN
- SVI (Switch Virtual Interface)
    - Can assign IP address
- DTP (Dynamic Trunking Protocol)
    - Only Cisco have
    - To determine whether port should be trunk or access
    - Have 2 modes
        - "dynamic auto"
        - "dynamic desirable"
- VTP (VLAN Trunking Protocol)
    - 3 modes
        - Server
        - Transparent
        - Client
    - To reset revision number
        - Change to transparent OR
        - Change to unused domain name

Commands

Switch

Interface

- "switchport trunk encapsulation dot1q"
- "switchport mode <mode>" (access, trunk)
- "switchport access vlan <vlan_num>
- "switchport trunk allowed vlan <vlan_num>
- "switchport trunk native vlan <vlan_num>
- "no switchport"
- "switchport nonegotiate"

Global config

- "ip routing"
- "default interface <interface>

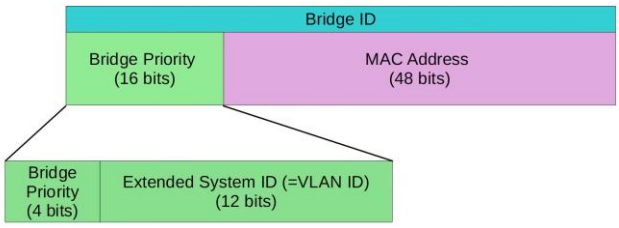
Router (sub interface)

- "encapsulation dot1q <vlan_num>

Spanning Tree (STP)

Normal STP

- Prevent broadcast storm
- MAC address flapping
- Ports have 3 states
    - **Designated**
    - **Root**
    - **Non-designated**
- Root bridge create BPDU
- Only designated port forward BPDU
- BPDU = Bridge Protocol Data Unit



Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

- Root Cost

Speed	STP Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

- **Port ID**
    - 128.<port num>
    - 128 is default, can change
- **Root Bridge**
    - **Lowest bridge ID**
    - **All designated port**
- **Root port**
    - **All switch other than root bridge have 1 root port**
    - **Lowest root cost**
    - **Lowest neighbour bridge ID**
    - **Lowest neighbour port ID**
- **Non-designated port**
    - **Switch with lower root cost**

- **Switch with lowest bridge ID**
- They will become designated, all others non-designated

- **STP States**

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Blocking	NO/YES	NO	NO	Stable
Listening	YES/YES	NO	NO	Transitional
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable
Disabled	NO/NO	NO	NO	Stable

- **Spanning Tree Timers**

STP Timer	Purpose	Duration
Hello	How often the root bridge sends hello BPDUs	2sec
Forward delay	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
Max Age	How long an interface will wait after ceasing to receive Hello BDPUs to change the STP topology.	20sec (10* hello)

- If never receive BPDU after max age, STP recalculate
- Timer on root bridge determine for the whole network
- **Ethernet Destination MAC address**
  - Cisco PVST+
    - 0100:0ccc:cccd
  - Standard STP
    - 0180:c2000:0000
- If set switch as root primary, priority = 24576, else will be 4096 lower than the lowest priority
- If set as secondary, priority = 28672

STP Optional Features

- **PortFast**
  - Straight away set to forwarding
  - Used on ports connected to end hosts
  - If receive BPDU, change to normal port

- **BPDU Guard**
  - Used on portfast interfaces
  - If receive BPDU, will be error-disabled
  - To recover
    - Shutdown, no shutdown
    - Auto feature - ErrDisable Recovery
      - Disabled by default
      - Default time 300s/5m

- **BPDU Filter**
  - Stop port from sending BPDU to end host
  - If receive BPDU
    - Global config - disable filter and portfast, act as normal
    - Interface - no effect
    - BPDU guard applied, filter applied in global config - error disabled

- **Root Guard**
  - Prevent a client switch from becoming a root port
  - Can only apply in interface, those connected to client network
  - If receive superior BPDU, become broken (root inconsistent) state
  - To re-enable
    - Stop receiving superior BPDU
    - Auto re-enable after 20s (BPDU max age)

- **Loop Guard**
  - Prevent loop caused by uni-directional link, usually caused by layer 1 issue
  - Normally in fibre optic
  - When max age timer reach 0, won't transition to designated port
    - Won't start sending BPDU
  - To re-enable
    - Auto when start receiving BPDU
  - Note, cannot apply both root and loop guard at the same time

Rapid Spanning Tree

- **Root cost**

Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	X	200
1 Tbps	X	20

- States

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable

- If administratively down - discarding

- **Differences**
  - Protocol Version

        - STP - 0
        - RSTP - 2
    - BPDU Type
        - STP - 0
        - RSTP - 2
    - BPDU Flags
        - STP uses 2 bits
        - RSTP uses all 8 bits

- **Port roles**
    - Designated and Root remain the same
    - Non-designated
        - Alternate port
            - Same as blocking state in STP
            - Alternate path towards root
            - Backup for root port
        - Backup port
            - Discarding port that receives a superior BPDU from another interface on the same switch
            - Occur because of hub, will not encounter nowadays
            - Backup for the designated port that it is receiving BDPU from
            - The higher priority will be the backup port
- **Inter-Operability**
    - Interface with RSTP connected to interface with STP will operate in STP mode
- RSTP the BPDU originate from all switches
- **Timer**
    - Max age = 6s
    - If never receive after max age, will remove MAC address learned on that interface
- **Link types**
    - Edge
        - Connected to end host
        - Straight away to forwarding state
        - Need to config
    - Point-to-point
        - Btw switches
        - Full duplex
        - Auto, don't need config
    - Shared
        - Connected to other switch via hub
        - Half duplex
        - Auto, don't need config
- **STP optional features built into RSTP**
    - UpLinkFast
    - BackboneFast
    - PortFast

Configs

- Basic STP
  - SW1# **show spanning-tree**
  - SW1(config)# **spanning-tree mode <mode>** (mst, pvst, rapid-pvst)
  - SW1(config)# **spanning-tree vlan <vlan_num> root primary/secondary**
  - SW1(config-if)# **spanning-tree vlan <vlan num> cost <cost>**
  - SW1(config-if)# **spanning-tree vlan <vlan num> port-priority <port-priority>**

- Additional Features

  PortFast
  - SW1(config-if)# **spanning-tree portfast [edge]**
  - SW1(config)# **spanning-tree portfast [edge] default**
  - SW1(config-if)# **spanning tree portfast trunk**

  BPDU Guard
  - SW1(config-if)# **spanning-tree bpduguard enable**
  - SW1(config)# **spanning-tree portfast [edge] bpduguard default**
  - SW1(config-if)# **spanning-tree bpduguard disable**
  - SW3(config)# **errdisable recovery cause bpduguard**
  - SW3(config)# **errdisable recovery interval <interval>**

  BPDU Filter
  - SW3(config-if)# **spanning-tree bpdu filter**
  - SW3(config)# **spanning-tree portfast [edge] bpdufilter default**
  - SW3(config-if)# **spanning-tree bpdufilter disable**

  Root Guard
  - SW2(config-if)# **spanning-tree guard root**

  Loop Guard
  - SW3(config-if)# **spanning-tree guard loop**
  - SW3(config)# **spanning-tree loopguard default**

- Rapid STP
  - SW1(config-if)# **spanning-tree portfast**
  - SW1(config-if)# **spanning-tree link-type point-to-point**
  - SW1(config-if)# **spanning-tree link-type shared**

EtherChannel

- EtherChannel also known as
  - LAG (Link Aggregation Group)

    - Port Channel

- 3 modes of EtherChannel config
    - PAgP (Port Aggregation Protocol)
    - LACP (Link Aggregation Control Protocol)
    - Static

- Up to 8 interface for EtherChannel
    - 16 for LACP (8 active, 8 standby)

- PAgP: "auto", "desirable"
- LACP: "active", "passive"
- Static: "on"

- Members must have the same matching configs
    - Same duplex
    - Same speed
    - Same switchport mode (access/trunk)
    - Same allowed VLANs/native VLAN (for trunk ports)
- If an interface don't match, it will be excluded from the EtherChannel
- Don't need to have the same group number

Config

- `ASW1# show etherchannel load-balance`
- `ASW1(config)# port-channel load-balance <method>`

- `ASW1(config-if-range)# channel-group <channel_num> mode <mode>`
- `ASW1(config-if-range)# channel-protocol <protocol> (LACP, PAgP)`

- `ASW1(config)# interface port-channel <channel_num>`
- `ASW1(config-if)# switchport trunk encpasulation dot1q`
- `ASW1(config-if)# switchport mode trunk`

- `ASW1(config-if-range)# no switchport`
- `ASW1(config-if-range)# channel-group <channel_num> mode <mode>`
- `ASW1(config)# interface port-channel <channel_num>`
- `ASW1(config-if)# ip address <ip_address> <netmask>`

- **"show etherchannel summary"**

Dynamic Routing

Intro

- When 1 link fail, static route remain unchanged
    - Dynamic routing find another path
- Router advertise routes they know
- Form 'adjacency' with neighbours to exchange info
- Use route with lowest metric

Types of Dynamic Routing Protocols

- IGP (Interior Gateway Protocol) - within 1 AS
    - Distance Vector
        - RIP (Routing Interface Protocol)
        - EIGRP (Enhanced Interior Gateway Routing Protocol)
    - Link State
        - OSPF (Open Shortest Path First)
        - IS-IS (Intermediate State - Intermediate State)
- EGP (Exterior Gateway Protocol) - multiple AS
    - Path Vector
        - BGP (Border Gateway Protocol)

Distance Vector

- Works by sending these to direct neighbours
    - Known destination networks
    - The corresponding metrics
- Called 'routing by rumour'

Link State Protocol

- Router create 'connectivity map'
- React faster to changes

Metric

- Lower metric = better
- ECMP: Equal Cost Multi-Path

IGP	Metric	Explanation
RIP	Hop count	Each router in the path counts as one 'hop'. The total metric is the total number of hops to the destination. **Links of all speeds are equal.**
EIGRP	Metric based on bandwidth & delay (by default)	Complex formula that can take into account many values. By default, the bandwidth of the **slowest link in the route** and the total delay of all links in the route are used.
OSPF	Cost	The cost of each link is calculated based on bandwidth. The total metric is the total cost of each link in the route.
IS-IS	Cost	The total metric is the total cost of each link in the route. The cost of each link is **not** automatically calculated by default. All links have a cost of 10 by default.

Route protocol/type	AD
Directly connected	0
Static	1
External BGP (eBGP)	20
EIGRP	90
IGRP	100
OSPF	110

Route protocol/type	AD
IS-IS	115
RIP	120
EIGRP (external)	170
Internal BGP (iBGP)	200
Unusable route	255

- Loopback = 1
- AD used before metric to choose route

Floating Static Route

- A static route with a higher AD
- Can be used as a backup route

RIP

- Industry standard
- Use hop count as metric
- Max hop count = 15
- 3 version
    - RIPv1 and v2 (IPv4)
    - RIPng (IPv6)
- 2 message type
    - Request
    - Response
- By default, share routing table every 30s
- RIPv1
    - Only advertise classful address, no VLSM/CIDR
    - No subnet mask info in advertisements
        - E.g. 10.1.1.0/24 -> 10.0.0.0 (assume /8 mask)
    - Message broadcast to 255.255.255.255
- RIPv2
    - Support CIDR/VLSM
    - Include subnet mask in advertisement
    - Message multicast to 224.0.0.9

EIGRP

- Cisco proprietary
- Faster reaction than RIP
- Must have the same AS group number btw different switches
- No 15 hop count limit
- Multicast to 224.0.0.10
- Can perform unequal cost load balancing

- Config
    - Uses wildcard mask
    - Router ID priority
        1. Manual config
        2. Highest IP address on loopback interface
        3. Highest IP address on physical interface

EIGRP Metric

- Metric = bandwidth of slowest link + delay of all links
- Default K values
    - K1, K3 = 1
    - K2, K4, K5 = 0

Terminology

Metric

- Feasible distance
- Reported distance (Advertised Distance)

Path

- Successor
- Feasible Successor
    - Alternate route that meets the feasibility condition

- Feasibility Condition
    - A route is considered a feasible successor if it's reported distance less than successor route's feasible distance

Unequal Cost Load Balancing

- Change the variance to allow it
- Feasible Successor routes with FD up to <variance-value> of Successor's route FD can be used

Commands

Configure RIP

- R1(config)# **router rip**
- R1(config-router)# **version <version_num>** (1,2)
- R1(config-router)# **no auto-summary**
- R1(config-router)# **network <ip address>**
    - Don't have netmask, is a classful address

Configure EIGRP

- R1(config)# **router eigrp**
- R1(config-router)# **no auto-summary**
- R1(config-router)# **network <ip address> <wildcard mask>**
- R1(config-router)# **passive-interface <interface>**
- R1(config-router)# **maximum-paths <max_path>**

- R1(config-router)# **distance <administrative distance>**
- R1(config-router)# **variance <variance>**
- R1(config-router)# **eigrp router-id <router_id>**
    - <router_id>: a.b.c.d

Configure Loopback Address

- R1(config)# **int loopback 1**
- R1(config-int)# **ip add <ip_add> <netmask>**
    - For loopback address, netmask: 255.255.255.255

Show commands

- "**show ip eigrp neighbors**"
- "**show ip route eigrp**"
- "**show ip protocols**"

OSPF

OSPF

- Open Shortest Path First
- Use Djikstra / Shortest Path First algorithm
- OSPFv1 old, v2 - IPv4, v3 -IPv6
- LSA: Link State Advertisement
- LSDB: Link State Database

LSA Flooding

- LSA default timer: 30mins
- LSA flooded again after timer expire

Process of sharing LSAs and determining best route

1. Become neighbours
2. Exchange LSAs
3. Calculate best routes

Areas

- Terms
    - Area
    - Backbone area
    - Internal router
    - Area Border Router (ABR)
    - Backbone Router
    - Intra-area route
    - Inter-area route

    - Autonomous System Boundary Router (ASBR)
- Rules
    - Areas must be contiguous
    - At least 1 ABR connected to Backbone area
    - Interfaces in same subnet must be in same area

Config

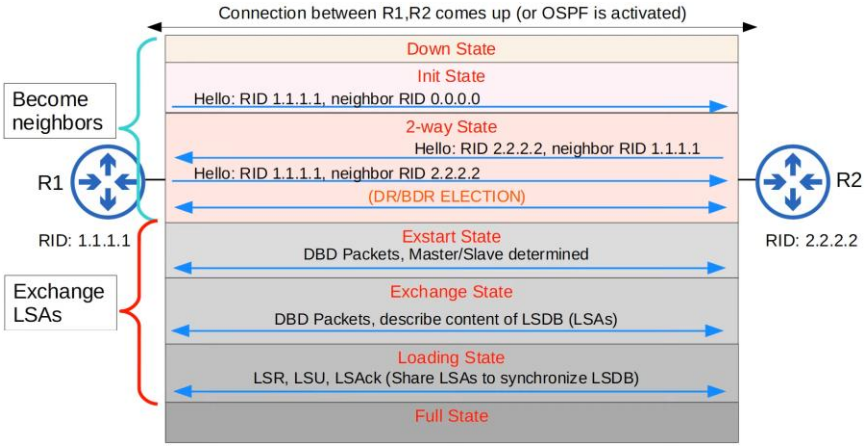
- Router ID order of priority
    1. Manual config
    2. Highest IP address on a loopback interface
    3. Highest IP address on a physical interface

OSPF cost

- OSPF metric is cost
- Cost = reference bandwidth / interface bandwidth
- Reference bandwidth default = 100mbps
- Min cost value = 1
- To change cost
    - Change from router ospf settings
    - Change from interface ospf cost
    - Change from interface bandwidth (not recommended)

OSPF Neighbours

- Hello timer (default = 10s on Ethernet)
- Hello message multicast to 224.0.0.5
- OSPF messages encapsulated in IP header
- IP header protocol field = 89
- Higher router ID = master



Type	Name	Purpose
1	**Hello**	Neighbor discovery and maintenance.
2	**Database Description (DBD)**	Summary of the LSDB of the router. Used to check if the LSDB of each router is the same.
3	**Link-State Request (LSR)**	Requests specific LSAs from the neighbor.
4	**Link-State Update (LSU)**	Sends specific LSAs to the neighbor.
5	**Link-State Acknowledgement (LSAck)**	Used to acknowledge that the router received a message.

Loopback Interface

- Provides reliable IP address that won't fail when interface fail
- Still can route to the router if an interface fail

OSPF Network Types

- Broadcast
    - Ethernet, FDDI (Fiber Distributed Data Interfaces)
- Point-to-Point
    - PPP(Point-to-Point), HDLC (High-Level Data Link Control)
- Non-broadcast
    - Frame relay, X.25

Broadcast Network Type

- Dynamically discover neighbours
- Sends/listen 'Hello' messages on multicast address 224.0.0.5
- DR and BDR (Designated Router) must have on each subnet
    - Others are DROther
    - If only 1 OSPF router, DR only
- DR/BDR selection
    - Highest OSPF interface priority (default = 1)
    - Highest OSPF router ID
- DR/BDR selection non pre-emptive
    - Once selected, won't change unless OSPF reset
- When reset
    - BDR -> DR
    - Select new BDR
    - Highest interface priority that was not DR/BDR will only become BDR
- 'Full' adjacency only btw DR/BDR
- '2-way' adjacency btw DROther and DROther
- Therefore, only exchange LSAs with DR/BDR
- DR/BDR are multicast using 224.0.0.6

Point-to-Point

- No DB/DBR
- Dynamically discover neighbours

- Send/listen 'Hello' messages with multicast address 224.0.0.5
- The 2 routers will from 'Full' adjacency
- Serial interface
    - Default encapsulation is HDLC
    - One side is DCE and other side is DTE
    - You must config the clk rate on the DCE side

Broadcast	Point-to-point
Default on **Ethernet**, **FDDI** interfaces	Default on **HDLC**, **PPP** (serial) interfaces
DR/DBR elected	No DR/BDR
Neighbors dynamically discovered	Neighbors dynamically discovered
Default timers: Hello 10, Dead 40	Default timers: Hello 10, Dead 40

(**Non-broadcast** network type default timers = Hello 30, Dead 120)

OSPF Neighbour Requirements

1. Area number must match
2. Interfaces must be in the same subnet
3. OSPF process must not be shutdown
4. OSPF router IDs must be unique
5. Hello and Dead timers must match
6. Authentication settings must match

The following rules still can become OSPF neighbours, but OSPF doesn't operate properly

1. IP MTU must match
2. OSPF network type must match

OSPF LSA Types

- 11 types, 3 for CCNA
    - Type 1 (Router LSA) - OSPF router
    - Type 2 (Network LSA) - DR
    - Type 5 (AS External LSA) - ASBR

Config

Config OSPF

- R1(config)# **router ospf <process ID>**
- R1(config-router)# **network <ip address> <wildcard mask> area <area>**
    1. "network 10.0.12.0 0.0.0.3"

    "network 10.0.23.0 0.0.0.3"

1. "network 0.0.0.0 255.255.255.255"

2. "network 10.0.0.0 0.0.255.255"
3. "network 10.0.12.1 0.0.0.0"
   "network 10.0.23.1 0.0.0.0"

- R1 (config-if)# **ip ospf <process-id> area <area-id>**

Passive Interface

- R1(config-router)# **passive-interface <interface>**
- R1 (config-router)# **passive-interface default**

Default route

- R1(config-router)#**default information originate**
  - Must set default route first
  - R1(config)# **ip route 0.0.0.0 0.0.0.0 <ip address>**

Router ID/ Admin Distance

- R1(config-router)# **router-id <router-id>**
- R1(config-router)# **distance <AD>**

Reference Bandwidth

- R1(config-router)# **auto-cost reference-bandwidth <speed in megabits-per-second>**
- R1(config-if)# **ip ospf cost <cost>**
- R1(config-int)# **bandwidth <bandwidth Kbps>**

OSPF Interface Priority

- R1(config-if)# **ip ospf priority <priority>**

Clear OSPF process

- R1# **clear ip ospf process**

Serial Connection

- R1(config-ifx)# **encapsulation ppp**
- R1# **show controllers <interface_id>**
- R1(config-if)# **clock rate <rate in bps>**

Authentication

- R1(config-if)# **ip ospf authentication-key** *password*
- R1(config-if)# **ip ospf authentication**

"show commands"

- "show ip ospf database"
- "show ip ospf neighbor"
- "show ip ospf interface"

- "show ip protocols"
- "show ip ospf int br"

FHRP

First Hop Redundancy Protocol

- Problem: Alternate route to Internet, but end host default gateway set to only 1 router
- Solution: FHRP, set default gateway as a virtual IP

How it works

- Routers send 'Hello' multicast to each other
    - Active & Backup chosen
- If PC send ARP Request to reach Internet, will get virtual MAC
- If Router fail
    - PC no change
    - SW need to change
        - since in MAC address table the MAC is connected to interface
        - Need change which interface connected to the MAC
- Backup router send **gratuitous ARP**
    - Broadcast w/o getting ARP request
- FHRPs are pre-emptive
    - Router won't give up role
- Default priority = 100

HSRP (Hot Standby Router Protocol)

- Cisco proprietary
- Version 2: support IPv6 and more groups can be configured
- Can config different active router in different subnet/VLAN to load balance

VRRP (Virtual Router Redundancy Protocol)

- Standard protocol
- Can config different master router

GLBP (Gateway Load Balancing Protocol)

- Cisco proprietary
- AVG (Active Virtual Gateway)
- 4 AVF (Active Virtual Forwarder) assigned by AVG
    - AVG can be AVF
- Each AVF is default gateway for portion of subnet

Table

FHRP	Terminology	Multicast IP	Virtual MAC	Cisco proprietary?
HSRP	Active/Standby	v1: 224.0.0.2 v2: 224.0.0.102	v1: 0000.0c07.acXX v2: 0000.0c9f.fXXX	Yes
VRRP	Master/Backup	224.0.0.18	0000.5e00.01XX	No
GLBP	AVG / AVF	224.0.0.102	0007.b400.XXYY	Yes

- X: group number
- Y: AVF number

Config (HSRP)

- R1(config-if)# **standby version 2**
- R1(config-if)# **standby <group-num> ip <virtual-ip>**
- R1(config-if)# **standby <group-num> priority <priority>**
- R1(config-if)# **standby <group-num> preempt**

TCP/UDP

Function of Layer 4 - Transport Layer

- Provide transparent transfer of data btw end hosts
- Provide services (or don't), e.g. reliable data transfer
- Provide Layer 4 addressing (port number)
    - Well known: 0 - 1023
    - Registered: 1024 - 49151
    - Ephemeral/private/dynamic: 49152 - 65535

Session Multiplexing

- PC1 send traffic to SRV1 over Internet
    1. PC1: TCP, Src-50000, Dst: 80 (HTTP)
        - PC1 -> SRV1
    2. SRV1: TCP, Src:80, Dst: 50000 (Opposite)
        - SRV1 -> PC1
- Can have multiple sessions
- The port numbers allow PC1 to know from which session

TCP

Establish Connection: 3-way Handshake

1. PC -> SRV (SYN)
2. SRV -> PC (ACK, SYN)
3. PC -> SRV (ACK)

Terminate Connection: 4-way handshake

1. PC -> SRV (FIN)
2. SRV -> PC (ACK)
3. SRV -> PC (FIN)
4. PC -> SRV (ACK)

Sequencing/Acknowledging

- Starting Sequence number random

3-way handshake

1. PC1 -> PC2 (Seq: 10)
2. PC2 -> PC1 (Seq:50. Ack: 11)
3. PC1 -> PC2 (Seq:11, Ack: 51)

Normal communication

1. PC2 -> PC1 (Seq:51, Ack:12)
2. PC1 -> PC2 (Seq:12, Ack:52)
3. PC2 -> PC1 (Seq:52, Ack:13)

Retransmission

1. PC1 -> PC2 (Seq:20)
2. PC2 -> PC1 (Ack:21)
3. PC1 -> X (Seq:21)
4. PC1 -> PC2 (Seq:21)
5. PC2 -> PC1 (Ack:22)

Flow Control: Window Size

1. PC1 -> PC2 (Seq:20)
2. PC1 -> PC2 (Seq:21)
3. PC1 -> PC2 (Seq:22)
4. PC2 -> PC1 (Ack:23)

- Sliding window
    - Adjust the window size dynamically

TCP	UDP
Connection-oriented	Connectionless
Reliable	Unreliable
Sequencing	No sequencing
Flow control	No flow control
Use for downloads, file sharing, etc	Used for VoIP, live video, etc

Port Numbers

TCP

- FTP Data: 20
- FTP Control: 21
- SSH: 22
- Telnet: 23
- SMTP: 25
- HTTP: 80
- POP3: 110
- HTTPS: 443

UDP

- DHCP server: 67
- DHCP client: 68
- TFTP: 69
- SNMP agent: 161
- SNMP manager: 162
- Syslog: 514

UDP & TCP

- DNS: 53

IPv6 Addressing

IPv6

- 128 bits

Shortening IPv6

- Remove leading 0s
- Consecutive quartets of 0s
    - ::

    - Can only have 1

(Modified) EUI-64

- Extended Unique Identifier
- Use MAC address
- Method
    - Divide MAC address in half
    - Insert FFFE in middle
    - Inverse 7th MSB
- Add this 'new MAC' together with the network part of IPv6 to get the full IPv6 address
- 2 types of MAC address
    - UAA (Universally Administered Address)
    - LAA(Locally Administered Address)
- 7th bit is U/L bit
    - 0: UAA
    - 1: LAA

Address Types

- Global unicast
- Unique local
- Link local
- Multicast
- Anycast
- Unspecified IPv6 addresses
- Loopback address

Global Unicast

- Public addresses that can be used over the Internet
- Range: any address that is not reserved
    - Originally: 2000::/3
- Network part
    - 48 bit: Global Routing prefix
    - 16 bit: Subnet Identifier
- Host part
    - 64 bit: interface identifier

Unique Local Address

- Private address that cannot be used over Internet
- Can be freely used over internal networks
- Don't need to be unique (global ID should be random, so if merge with other companies, won't be the same)
- Range: FC::/7
- Network
    - 8 bit: FD, indicates Unique Local address
    - 40 bit: global ID

    - 16 bit: subnet identifier
- Host
    - 64 bit: interface identifier

Link Local Address

- Auto generated on IPv6 enabled interface
- FE80::/10
    - 54 bits after FE80/10 is 0, so only FE80, no FE9 etc
- Interface ID generated using EUI-64 rules
- Used within a single subnet, routers won't route it
- Used for
    - Routing protocols peerings (OSPFv3 uses it for adjacencies)
    - Next-hop address for static routes
    - Neighbour Discovery Protocol (ARP for IPv6)

Multicast Address

- FF00::/8
- No broadcast

Purpose	IPv6 Address	IPv4 Address
All nodes/hosts (functions like broadcast)	FF02::1	224.0.0.1
All routers	FF02::2	224.0.0.2
All OSPF routers	FF02::5	224.0.0.5
All OSPF DRs/BDRs	FF02::6	224.0.0.6
All RIP routers	FF02::9	224.0.0.9
All EIGRP routers	FF02::A	224.0.0.10

- Address scopes (how far packet will be forwarded)
    - Interface local (FF01): within device
    - Link Local (FF02): subnet
    - Site-local (FF05): within physical location
    - Organization level (FF08): wider than site local
    - Global (FF0E): routed over internet

Anycast Address

- 1 to 1-of-many
- Routers configured with same address
- No specific address range, can be anything

Unspecified IPv6 address

- :: - all 0s
- When device doesn't know it's IPv6 address
- Default route ::/0

Loopback Address

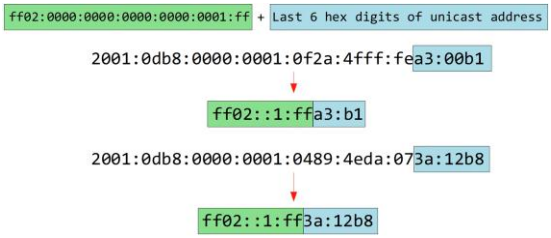
- ::1

IPv6 Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
12	96																																
16	128																																
20	160																																
24	192	Destination Address																															
28	224																																
32	256																																
36	288																																

- Fixed size: 40B
- Version: 0x0110 (6)
- Traffic class: QoS
- Flow label: Identify specific 'traffic flows'
- Payload length
- Next header: protocol
- Hop limit
- Source
- Destination

Solicited-Node Multicast Address



Neighbour Discovery Protocol (NDP)

- Work like ARP for IPv6
- See notes for relation btw solicited-node multicast address
- 2 messages
    - Neighbour Solicitation (NS) - ICMPv6 type 135
        - R1 -> R2
        - Src IP: R1 G0/0 IP
        - Dst IP: R2 solicited-node multicast address
        - Src MAC: R1 G0/0 MAC
        - Dst MAC: Multicast MAC based on R2's solicited-node address
    - Neighbour Advertisement (NA) - ICMPv6 type 136
        - R2 -> R1
        - Src IP: R2 G0/0 IP

        - Dst IP: R1 G0/0 IP
        - Src MAC: R2 G0/0 MAC
        - Dst MAC: R1 G0/0 MAC

- Function 2: auto discover routers
- 2 messages
    - Router Solicitation (RS)
        - ICMPv6 type 133
        - Multicast address FF00::2 (all routers)
        - "Is there any routers?"
    - Router Advertisement (RA)
        - ICMPv6 type 134
        - Multicast Address FF02::1 (all nodes)
        - "I am a router"

Stateless Address Auto-Configuration (SLAAC)

- Uses RS/RA to learn IPv6 prefix of local link
- Uses EUI-64 to generate interface ID

Duplicate Address Detection (DAD)

- Check if other devices on local link have same address
- 2 messages
    - NS
    - NA
- Host send itself NS
    - If receive reply, another host has same address
    - Else, address is unique

IPv6 Static Routing

- Process is separate from ipv4 routing
- Must be enabled
    - Else, cannot route them
- Types of route
    - Directly attached static route: only interface, ethernet cannot
    - Recursive static route: only ip
    - Fully specified: both ip and interface

Config

Enable IPv6 routing

- R1(config)# **ipv6 unicast-routing**

Set IP address

- R1(config-if)#**ipv6 route** *destination/prefix-length* {*next-hop* | *exit-interface* [*next-hop*]} [*ad*]
- R1(config-if)# **ipv6 <ipv6 address>/<subnet mask>**

- R1(config-if)# **ipv6 <ipv6 address>/<subnet mask> eui-64**
- R1(config-if)# **ipv6 <ipv6 address>/<subnet mask> anycast**
- R1(config-if)#**ipv6 address autoconfig**

Enable IPv6 on interface (enable link local)

- R1(config-if)# **ipv6 enable**

Show

- "show ipv6 neighbour"
- "show ipv6 interface brief"

ACL

ACL

- **Access Control List**
- Filter packets based on
    - Source/Destination IP
    - Source/Destination Ports
    - Etc

- They are an ordered sequence of ACEs (Access Control Entries)
- ACL must be applied to an interface
    - Inbound
    - Outbound
- 1 interface max
    - 1 outbound
    - 1 inbound
- If packet match an ACE, ignore rest of ACEs in the ACL
- ACEs processed from top to bottom
- Implicit deny - have in every end of ACL
    - Packet will be dropped if never meet any ACEs

ACL Types

Standard ACLs

- Match based on Source IP address only
- Types
    - Standard Numbered ACLs: **1-99, 1300-1999**
    - Standard Named ACLs
- Apply ACL close to the destination

Extended ACLs

- Match based on Source/Destination IP, Source/Destination Port, etc
- Type
    - Extended Numbered ACLs: **100-199, 2000-2699**
    - Extended Named ACLs
- Apply ACL close to the source

Configure Numbered ACLs same as Named ACLs

- Advantage
    - Can easily delete individual entries
        - If apply in global config, need to delete the whole ACL
    - Can insert new entries btw other entries by specifying sequence number

Resequencing ACLs

- Change the sequence number of the ACEs
- Allow to place entries in btw if sequence previously (1,2,3,4,5)

Protocols

- ICMP: 1
- TCP: 6
- UDP: 17
- EIGRP: 88
- OSPF: 89

Commands

Standard Numbered ACL

- R1(config)# **access-list** *number* {**deny** | **permit**} *ip-address wildcard-mask*
    - R1(config)# **access-list** *number* {**deny** | **permit**} *ip-address* (for /32)
    - R1(config)# **access-list** *number* {**deny** | **permit**} **host** *ip-address* (for /32)
    - R1(config)# **access-list** *number* **permit any**
- R1(config)# **access-list** *number* **remark** *remark*
- R1(config-if)# **ip access-group** *number* {**in** | **out**}

Standard Named ACLs

- R1(config)# **ip access-list standard** *acl-name*
- R1(config-std-nacl)# [*entry-number*] {**deny** | **permit**} *ip-address wildcard-mask*

Extended ACLs

- R1(config)# **ip access-list extended** {*name* | *number*}
- R1(config-ext-acl)# [*seq-num*] [**permit** | **deny**] *protocol src-ip dst-ip*
- R1(config-ext-nacl)# **deny tcp** *src*-ip [**eq** *src-port-num*] *dest-ip* [**eq** *dst-port-num*]
    - "eq","gt", "lt", "neq"
    - "range 80 100"

Delete Entry

- R1(config-std-nacl)# **no** *sequence-number*

Re-sequence entries

- R1(config)# **ip access-list** *resequence acl-id starting-seq-num increment*

Show

- "show access-lists"
- "show ip access-lists"
- "show running-config | include access-list"
- "show running-config | section access-list"

CDP & LLDP

Layer 2 Discovery Protocol

- Share info and discover connected neighbours
- Not recommended since security risk

Cisco Discovery Protocol (CDP)

- Enabled by default
- Multicast: **0100.0CCC.CCCC**
- When receive CDP message
    - Process and discard
    - Interface must be up
- Default timers
    - Sent: **60s**
    - Hold time: **180s**
- CDPv2 by default

Link Layer Discovery Protocol (LLDP)

- IEEE 802.1AB
- Disabled on Cisco device
- Can run both LLDP and CDP at the same time
- Multicast: **0180.C200.000E**
- When receive LLDP message
    - Process and discard
- Default timers
    - Sent: **30s**
    - Hold time: **120s**
    - Re-initialization: **2s**

- Both protocols can run together

Config

General

- R1(config)# {**lldp/cdp**} **run**
- R1(config-if)# **cdp enable**
    - R1(config-if)# **lldp transmit**
    - R1(config-if)# **lldp receive**
- R1(config)# {**lldp/cdp**} **timer** *seconds*
- R1(config)# {**lldp/cdp**} **holdtime** *seconds*
- R1(config)# {**lldp/cdp**} **reinit** *seconds*

Show

- R1# **show** {**lldp/cdp**}
- R1# **show** {**lldp/cdp**} **traffic**
- R1# **show** {**lldp/cdp**} **interface**
- R1# **show** {**lldp/cdp**} **neighbors**
- R1# **show** {**lldp/cdp**} **neighbors detail**
- R1# **show** {**lldp/cdp**} **entry** *name*

NTP

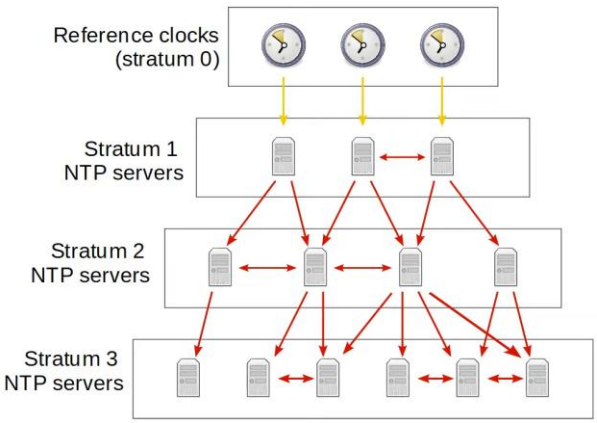
Clock

- Used for logging and troubleshooting
- 2 types of clock
    - Hardware (Calendar)
    - Software (Clock)

NTP (Network Time Protocol)

- Auto set clock
- NTP client request from NTP server
- Can be client and server simultaneously
- Accuracy
    - Within 1ms in LAN
    - Within 50ms over WAN/Internet
- UDP port 123

NTP Hierarchy



- Max stratum = 15
- 3 NTP modes
    - Server
    - Client
    - Symmetric Active (Peer)
- Can be all 3 modes simultaneously
- 2 server types
    - Primary: connected to reference clock
    - Secondary: connected to NTP servers

Config

Basic Clock

Set clock

- R1# **clock set** *hh:mm:ss {day I month} {month I day} year*
- R1# **calendar set** *hh:mm:ss {day I month} {month I day} year*

Sync clock

- R1# **clock update-calendar**
- R1# **clock read-calendar**

Set time-zone/ daylight saving

- R1(config)# **clock timezone** *name hours-offset [minutes-offset]*
- R1(config)# **clock summer-time recurring** *name start end [offset]*

NTP

Basic Configuration Commands

- R1(config)# **ntp server** *ip-address* [**prefer**]
- R1(config)# **ntp peer** *ip-address*
- R1(config)# **ntp update-calendar**
- R1(config)# **ntp master** [*stratum*]
- R1(config)# **ntp source** *interface*

Basic Authentication Commands

- R1(config)# **ntp authenticate**
- R1(config)# **ntp authentication-key** *key-number* **md5** *key*
- R1(config)# **ntp trusted-key** *key-number*

Below for client only

- R1(config)# **ntp server** *ip-address* **key** *key-number*
- R1(config)# **ntp peer** *ip-address* **key** *key-number*

Basic Show Commands

- R1# **show ntp associations**
- R1# **show ntp status**

DNS

Purpose of DNS

- Resolve names to IP addresses
- Easier to remember names
- DHCP is used to auto learn DNS server

How it works

- PC send DNS request (e.g. ping youtube.com)
- Router sends the packets as normal traffic to its set route for the DNS server
- Request reach DNS server
- DNS server reply with the IP address
    - 'A' record - IPv4
    - 'AAAA' record - IPv6
- Typical DNS queries/response is UDP
    - If message > 512 bytes, TCP
    - Port number 53 for both TCP/UDP

Cisco Config

- Don't need do anything as router forward DNS messages as usual
- Can set routers as
    - DNS server
    - DNS client

Config

Set as router

- R1(config)# **ip dns server**
- R1(config)# **ip host** *name ip-address*
- R1(config)# **ip name-server** *ip-address*

- R1(config)# **ip domain lookup**

Set as client

- R1(config)# **ip name-server** *ip-address*
- R1(config)# **ip domain lookup**
- R1(config)# **ip domain name** *domain-name*
    - Look at notes for more info

Show

- "show hosts"

Windows

- "**ipconfig /all**"
- "**nslookup** *name*"
- "**ipconfig /displaydns**"
- "**ipconfig /flushdns**"
- "**ping** *ip-address* **-n** *number*"
    - "number" - number of pings sent

DHCP

DHCP (Dynamic Host Configuration Protocol)

- Allow hosts to automatically learn aspects of their network config
- Normally for client devices like PC and handphone
    - Network devices like routers use static config

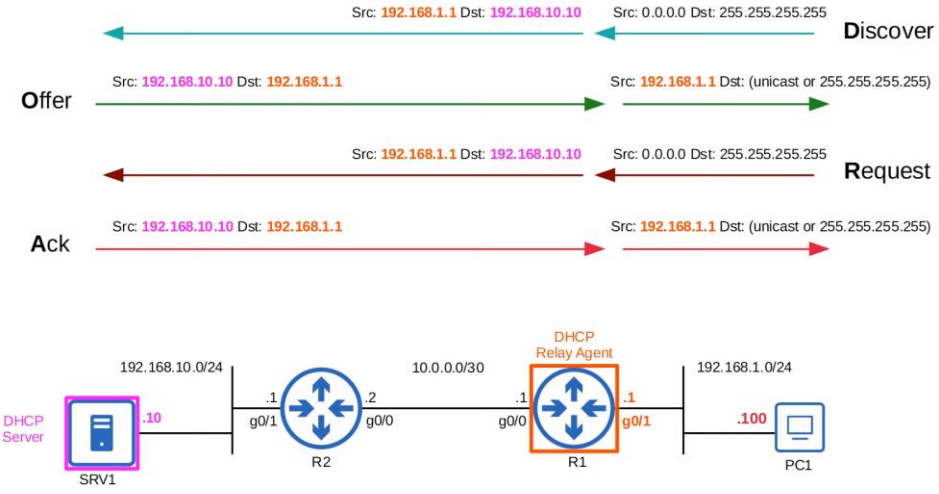
Discover	Client → Server	Broadcast
Offer	Server →Client	Broadcast or Unicast
Request	Client → Server	Broadcast
Ack	Server →Client	Broadcast or Unicast

Release	Client → Server	Unicast

- Bootp flags used to indicate broadcast or unicast
  - Decided by client
- Port
  - DHCP Server: UDP 67
  - DHCP Client: UDP 68

DHCP relay

- Router will relay DHCP messages from client to DHCP server



Config

DHCP Server

- R1(config)# **ip dhcp excluded-address** *ip-address-start ip-address-end*
  - Range of addresses
- R1(config)# **ip dhcp pool** *pool-name*
- R1(dhcp-config)# **network** *ip-address {network-mask | /prefix-length}*
- R1(dhcp-config)# **dns-server** *ip-address*

- R1(dhcp-config)# **domain-name** *domain-name*
- R1(dhcp-config)# **default-router** *ip-address*
- R1(dhcp-config)# **lease** {*days hours minutes* | **infinite**}

DHCP Relay Agent

- R1(config)# **interface** *interface*
    - Interface facing the clients
- R1(config-if)# **ip helper-address** *server-ip-address*

DHCP Client

- R1(config-if)# **ip address dhcp**

Show

- R1# **show ip dhcp binding**
- R1# **show ip interface** *interface*

PC

- "**ipconfig /release**"
- "**ipconfig /renew**"

SNMP

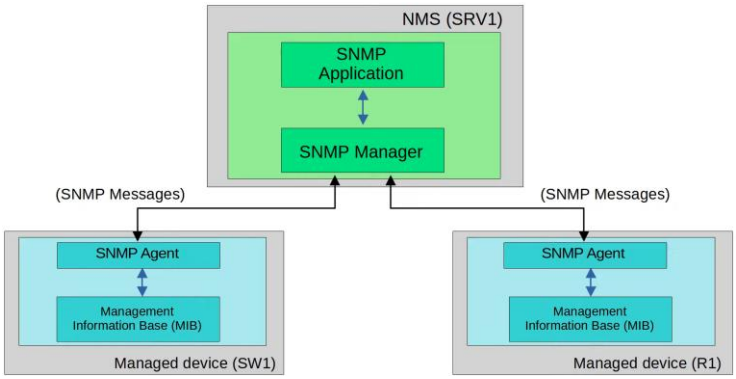
Overview

- SNMP (Simple Network Management Protocol)
- Industry standard
- 2 main devices
    - Managed device
    - Network Management Station (NMS)

Operations

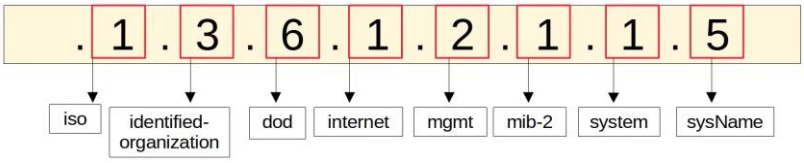
- There are 3 main operations
    - Managed devices notify NMS of events
    - NMS ask managed device for info
    - NMS tell managed device to modify config

Components



- NMS
    - Manager: software
    - Application: interface
- Managed device
    - Agent: software
    - MIB: structure that contains variables, identified by Object ID (OID)

Object ID (OID)



SNMP Versions

- SNMPv1
    - Original
- SNMPv2c
    - Can retrieve more info
    - 'c' - community strings, password in v1, removed in v2, added in v2c
- SNMPv3
    - More secure, best to use

SNMP Messages

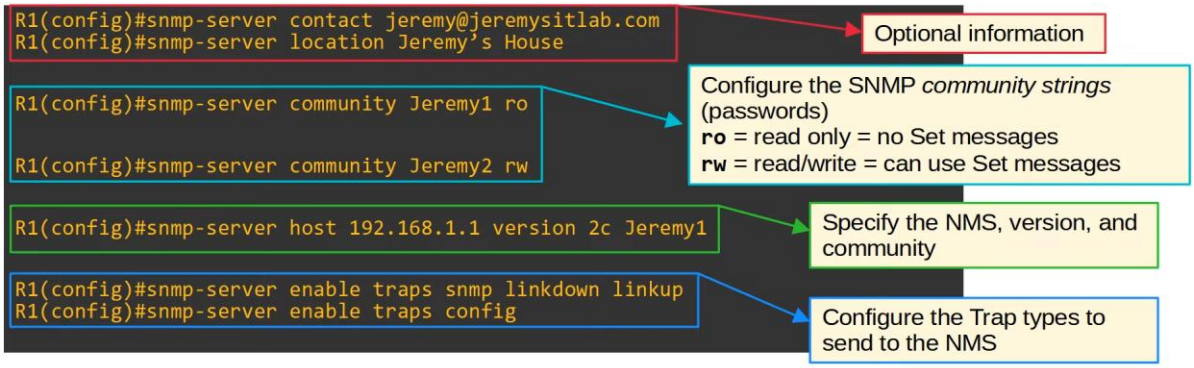
Message Class	Description	Messages
Read	Messages sent by the **NMS** to read information from the **managed devices**. (ie. What's your current CPU usage %?)	*Get* *GetNext* *GetBulk*
Write	Messages sent by the **NMS** to change information on the **managed devices.** (ie. change an IP address)	*Set*
Notification	Messages sent by the **managed devices** to alert the **NMS** of a particular event. (ie. interface going down)	*Trap* *Inform*
Response	Messages sent in response to a previous message/request.	*Response*

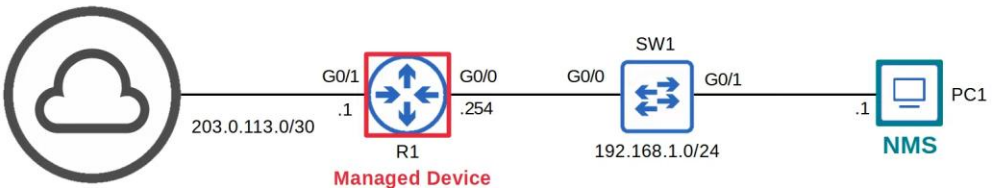
- Read
  - GetNext / GetBulk: get next element of MIB, bulk is better version
- Notification
  - Trap: no response
  - Inform: have response
- All have response except trap

Port numbers

- Agent = UDP 161
- Manager = UDP 162

Config





Syslog

Overview

- Industry standard
- Used to log events

Format

```
seq: time-stamp:  %facility-severity-MNEMONIC: description
```

Severity Level

Level	Keyword	Description
0	**Emergency**	System is unusable
1	**Alert**	Action must be taken immediately
2	**Critical**	Critical conditions
3	**Error**	Error conditions
4	**Warning**	Warning conditions
5	**Notice**	Normal but significant condition (**Notification**)
6	**Informational**	Informational messages
7	**Debugging**	Debug-level messages

- Every Awesome Cisco Engineer Will Need Ice Cream Daily

Logging Locations

- Console line
  - Default
- VTY lines
  - Telnet/SSH
- Buffer
  - RAM
  - Default
- External server
  - Ext server will listen at UDP 514

Config

Logging to console line

- R1(config)# **logging console** *severity*

Logging to VTY line

- R1(config)# **logging monitor** *severity*
- R1# **terminal monitor**

Logging to buffer

- R1(config)# **logging buffered** [*size*] *severity*

Logging to external server (both the same)

- R1(config)# **logging** *server-ip*
- R1(config)# **logging host** *server-ip*
- R1(config)# **logging trap** *severity*

New line if command interrupted

- R1(config)# **line console** *line*
- R1(config-line)# **logging synchronous**

Others

- R1(config)# **service timestamps log [datetime | uptime]**
- R1(config)# **service sequence-numbers**

SSH & Telnet

SSH

- Secure Shell
- Port: TCP 22
- "exec-timeout": after inactivity of set time, logout
- Layer 2 switch
    - Need default gateway so that PCs not in the VLAN/subnet can access the network device

Telnet

- Teletype Network
- No encryption, send data in plain text

SSH

- SSHv2 more secure than SSHv1
- If run both version, version 1.99
- Have data encryption and authentication
- IOS image will have K9 if support SSH
- NPE (No Payload Encryption) IOS image
    - Used if don't want device to have encryption
- FQDN (Fully Qualified Domain Name)
    - Host name + Domain Name
    - Used to generate the RSA key
- To generate RSA key
    - Must have hostname and domain name

Config

Login (password only)

- R1(config)# **line console 0**
- R1(config-line)# **{password | secret}** *password*
- R1(config-line)# **login**

Login Local (username and password)

- R1(config)# **username** *username* **secret** *password*
- R1(config)# **line console 0**
- R1(config-line)# **login local**

Layer 2 management switch

- SW1(config)# **interface vlan** *vlan-no*
- SW1(config-if)# **ip address** *ip-address subnet-mask*
- SW1(config-if)# **no shutdown**
- SW1(config)# **ip default-gateway** *ip-address*

Telnet

- SW1(config)# **enable secret** *password*
- SW1(config)# **username** *username* **secret** *password*

SSH

- SW1(config)# **hostname** *name*
- SW1(config)# **ip domain-name** *domain-name*
- SW1(config)# **crypto key generate ssh**
- SW1(config)# **enable secret** *password*
- SW1(config)# **username** *username* **secret** *password*
- SW1(config)# **ip ssh version 2**

Telnet/SSH

- SW1(config)# l**ine vty 0 15**
- SW1(config-line)# **login local**
- SW1(config-line)# **exec-timeout** *min sec*
- SW1(config-line)# **transport {input | output}** *type*
- SW1(config-line)# **access-class** *ACL-num* **{in | out}**

Show

- SW1# **show version**
- SW1# **show ip ssh**

PC

- "**telnet** *ip-address*"
- "**ssh -l** *username ip-address*"
- "**ssh** *username***@***ip-address*"

FTP & TFTP

Overview

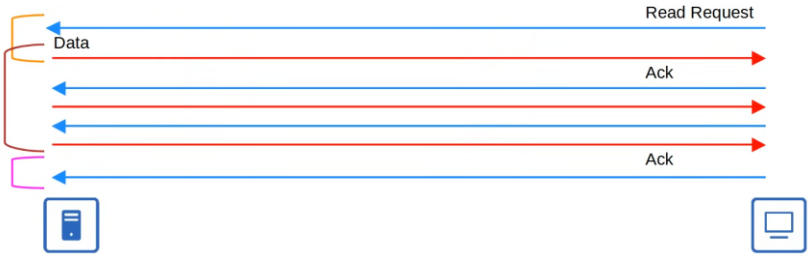
- FTP: File Transfer Protocol
- TFTP: Trivial FTP
- Industry standard
- Client-server model

TFTP

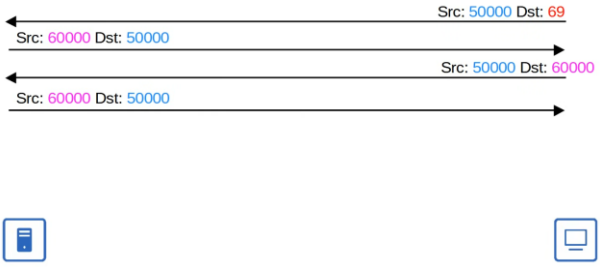
- Can only copy file to/from a server
- Does not replace FTP
- Used when simplicity more important than functionality
- TFTP servers listen at UDP port 69

Reliability

- Every TFTP message is acknowledged
- If never receive message after set time, will retransmit
- Only send 1 message at time (lock-step communication)
- File transfer have 3 phases
    1. Connection
    2. Data transfer
    3. Connection Termination



- Transfer Identifier (TID)
    - Port 69 only used to establish communication at the start



FTP

- FTP data: TCP port 20
- FTP control: TCP port 21
- Username and password used, however no encryption
    - Data still in plain text
- For greater security

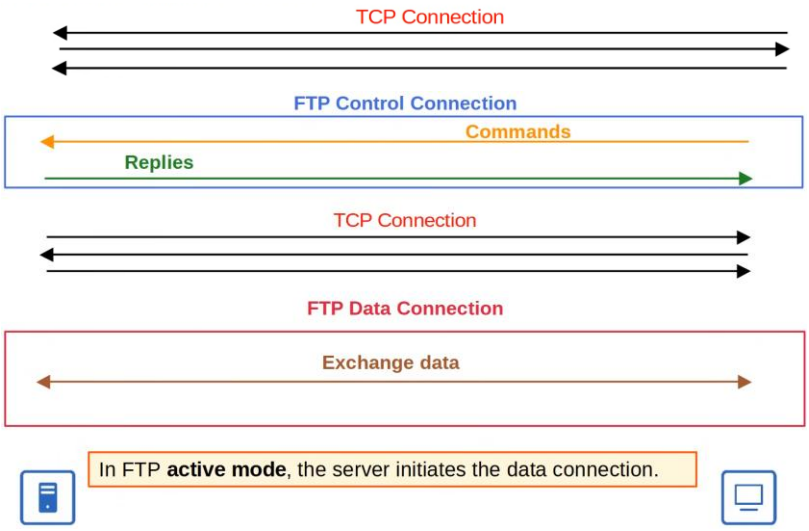
    - FTPS (FTP over SSL/TLS), upgrade to FTP
    - SFTP (SSH FTP), new protocol
- More features
    - Copy files
    - Navigate file directories
    - Add/remove directories
    - List files
- Client send FTP commands to perform these functions

Connections

- 2 types of connections
    - FTP control connection (TCP 21)
    - FTP data connection (TCP 20)

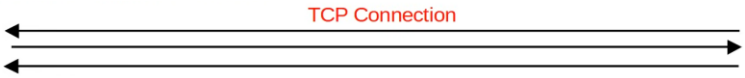
Active mode

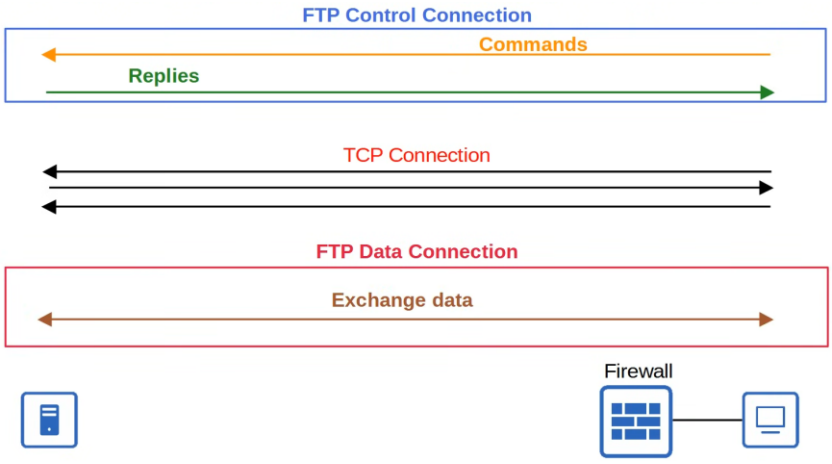
- Control connection is not terminated, there are 2 connection throughout the session
- Active mode, server initiate data connection



Passive mode

- Passive mode, client initiate data connection
- Usually for devices behind firewall as firewall usually don't allow outside devices to establish a connections





- Note: 1 TCP handshake for control connection, another for data connection

TCP vs UDP

FTP	TFTP
-Uses TCP (20 for data, 21 for control) for connection-based communication	-Uses UDP (69) for connectionless communication (although a basic form of 'connection' is used within the protocol itself)
-Clients can use FTP commands to perform various actions, not just copy files	-Clients can only copy files to or from the server
-Username/PW authentication	-No authentication
-More complex	-Simpler

IOS file system

- Way of controlling how data is stored and received
- "disk" type
  - Storage devices such as flash memory
- "opaque" type
  - Used for internal functions
- "nvram" type
  - Internal NVRAM, startup-config file is stored here
- "network"
  - Represents external file systems, for example, external FTP/TFTP servers

Config

FTP: copy files

- R1(config)# **ip ftp username** *username*
- R1(config)# **ip ftp password** *password*
- R1# **copy ftp: flash:**
- Address or name of remote host []? *Ftp-server-ip-address*
- Source filename []? *Source-filename*

- Destination filename [default]? *File-name*
    - Name of file you want to save as in file

TFTP: upgrade IOS

- R1# **copy tftp: flash:**
- Same as above
- R1(config)# **boot system** *file-path*
    - "*file-path*": **flash**:*filename*
- R1# **write memory**
- R1# **reload**
- R1# **delete** *file-path*

Show

- R1# **show flash**
- R1# **show file systems**
- R1# **show version**

NAT

Private IPv4 address

- One of the ways to provide enough IPv4 addresses to the world
- IPv4 Address range (RFC 1918)
    - 10.0.0.0/8
    - 172.16.0.0/12 (till 172.31.255.255)
    - 192.168.0.0/16
- Private IP cannot be used in Internet

NAT

- Network Address Translation
- Allow hosts with private address to communicate over Internet

Static NAT

- 1-to-1 mappings of private address to public address
- Map inside local to inside global address
- Cannot have the same inside global address

Dynamic NAT

- Same as NAT, but auto configure the address instead of manually
- ACL used to identify which traffic should be translated
    - If don't match, traffic NOT dropped, but not translated
- If not enough inside global address

    - Router drop packet
    - Can wait for timeout / manually clear NAT entries

PAT (NAT overload)

- Translate port addresses (if necessary)
- Allow hosts to use the same inside global address (router interface IP address)
- Port number will only be changed if 2 hosts uses the same port number

Terms

- Inside/Outside = Location of host
- Local/Global = Perspective

Config

- R1(config-if)# **ip nat {inside | outside}**

Static NAT

- R1(config)# **ip nat inside source static** *inside-local-ip inside-global-id*

Dynamic NAT

- R1(config)# **access-list** *number* **permit** *ip-address subnet-mask*
- R1(config)# **ip nat pool** *pool-name ip-start ip-end prefix-length {subnet-mask | prefix}*
- R1(config)# **ip nat inside source list** num **pool** *pool-name*

PAT (pool)

- Same as dynamic NAT except last command, add '**overload**'

PAT (interface)

- Use the outside interface IP address
- R1(config)# **access-list** *number* **permit** *ip-address subnet-mask*
- R1(config)# **ip nat inside source list** *number* **interface** *interface* **overload**

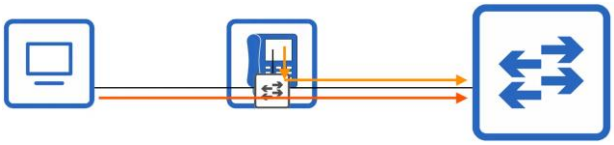
Show

- R1# **show ip nat translations**
- R1# **show ip nat statistics**
- R1# **clear ip nat translations ***

QoS

IP Phones

- Traditional phone operate over Public Switched Telephone Network (PSTN)
    - Also known as Plain Old Telephone Service (POTS)
- Use VoIP (Voice over IP)
    - Enable phone calls over IP network like Internet
- Have an internal 3 port switch
    - Uplink to external switch
    - Downlink to PC
    - Internally connected
- Allow PC and phone to use 1 switchport on the external switch
- Use Voice VLAN to separate phone traffic
    - External switch will use CDP to tell the phone to tag it's traffic in the voice VLAN
    - External switch interface = access port / untagged port



PoE (Power over Ethernet)

- Allows Power Sourcing Equipment (PSE) to provide power to Powered Devices (PD) over Ethernet
- Normal low power items
- Same cables used to provide data
- Normally PSE is a switch, convert AC to DC, send DC to devices
- Process to determine how much power to give
    - PSE first send low power signal and monitor response
    - If PD need to boot, PSE supply power
    - PSE continue monitoring and supplying as needed
- Power policing: prevent PD too much power
    - "power inline police" / "power inline police action err-disable"
        - Enable default settings - disable port, syslog message
        - "shutdown" "no shutdown" to re-enable
    - "power inline action log"
        - Restart interface which resets power negotiation

Name	Standard #	Watts	Powered Wire Pairs
Cisco Inline Power (ILP)	Made by Cisco, not standard	7	2
PoE (Type 1)	802.3af	15	2
PoE+ (Type 2)	802.3at	30	2
UPoE (Type 3)	802.3bt	60	4
UPoE+ (Type 4)	802.3bt	100	4

Classification

- Organize network traffic (packet) into traffic classes (categories)
  - ACL
  - NBAR (Network Based Application Recognition)
  - Layer 2/3 headers
- Headers contain fields to classify data
  - PCP in 802.1Q tag
  - DSCP in IPv4

PCP

- Also known as CoS (Class of Service)
- IEEE 802.1q
- 3 bits = 8 values

PCP value	Traffic types
0	Best effort (default)
1	Background
2	Excellent effort
3	Critical applications
4	Video
5	Voice
6	Internetwork control
7	Network control

- 0, lowest priority, no guarantee of meeting QoS standard
- IP phones mark call signalling traffic as PCP3, actual voice as PCP5
- PCP only found in
  - Trunk links
  - Access links with voice VLAN

IP Type of Service (TOS) byte

- Old
    - IP Precedence (IPP) - 3 bits
    - Other purposes - 5 bits
- New
    - DSCP - 6 bits
    - ECN - 2 bits

IPP

- Markings
    - 6,7: Reserved
    - 5: voice
    - 4: video
    - 3: voice signalling
    - 0: best effort

DSCP

- Markings
    - Default Forwarding (DF) - best effort traffic
    - Expedited Forwarding (EF) - low loss/latency/jitter traffic (usually voice)
    - Assured Forwarding (AF) - 12 standard values
    - Class Selector (CS) - 8 standard values, provides compatibility with IPP

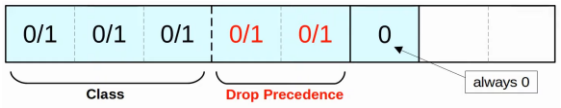
DF (Default Forwarding)

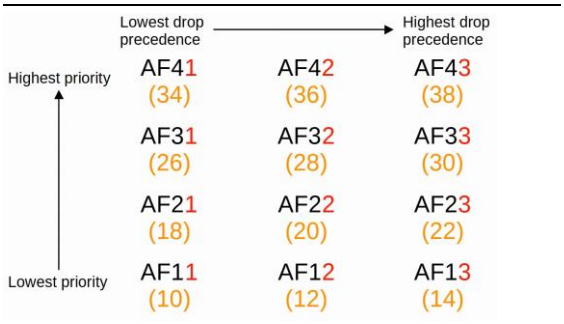
- Format: 000 000 XX
- DSCP value = 0
- Best effort traffic

EF (Expedited Forwarding)

- Format: 101 110 XX
- DSCP value = 46
- Traffic that require low loss etc

AF (Assured Forwarding)

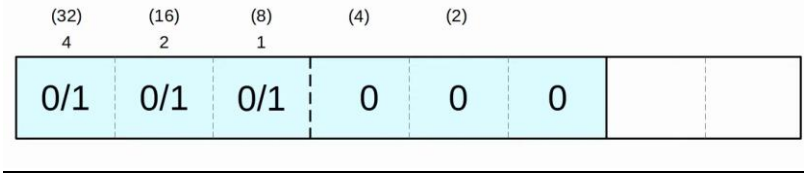




- 4 class, same class = same priority
- 3 level of drop precedence, higher = more likely drop

- Format: AFXY (X: class, Y: drop)
- DSCP value = 8X + 2Y

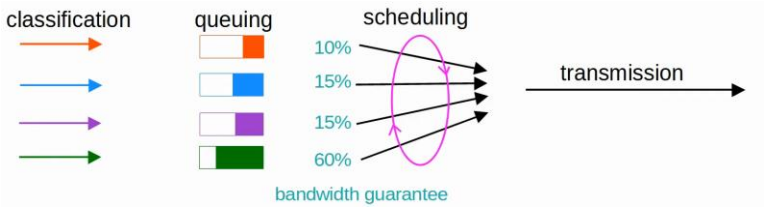
CS (Class Sector)



- Compatible with IPP
- Format: CSX (X = 0 - 7)
- DSCP value = 8X

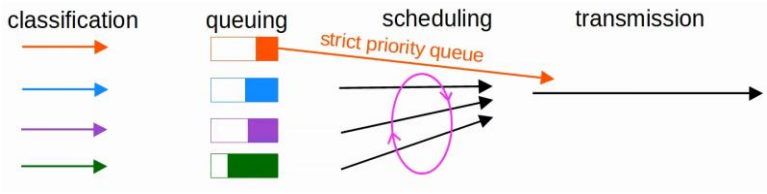
Queueing

- QoS uses multiple queues
- Scheduler used to decide which queue to forward traffic
- Methods
  - Weighted round robin
    - Each take equal turn, for specific queues, can get more
  - **CBWFQ (Class-Based Weighted Fair Queueing)**



- Weighted round-robin, with guaranteed bandwidth during congestion
- Bad: since take turn and wait for other queues, not good for voice/video

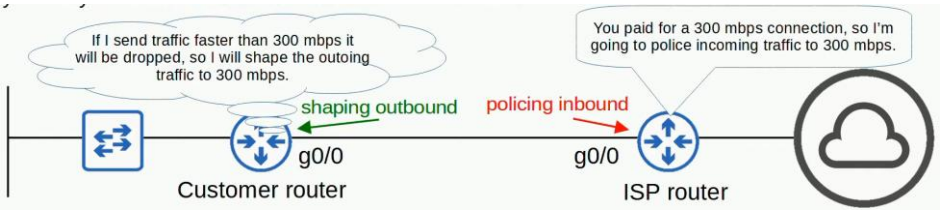
- **LLQ (Low Latency Queue)**



- 1 or more **strict priority queue**
- If there is traffic in the **strict priority queue**, WILL finish the queue before moving to next one
- Bad: Starvation
- Policing can control how much taken from that queue

Shaping and Policing

- **Shaping**: buffer traffic in a queue if outbound < inbound
- **Policing**: drop traffic if over configured rate
  - Burst - Other option of re-marking
  - Burst traffic over configured rate allowed for short times



PHB (Per-Hop Behaviour)

- QoS is unique only to a particular device
- E.g. R1 may classify HTTP packets as high priority, but when goes to R2, it will be based on R2 config

Config

Voice VLAN

- SW1(config)# **switchport mode access**
- SW1(config-if)# **switchport voice vlan** *vlan-num*

Power Policing

- SW1(config-if)# **power inline police** [**action** {**err-disable** | **log**}]

QoS

1. What kind of traffic you want to filter
    - R1(config)# **class-map** *map-name*
    - R1(config-cmap)# **match protocol** *protocol*

2. What kind of treatment you want to give
    - R1(config)# **policy-map** *map-name*
    - R1(config-pmap)# **class** *class-map-name*
    - R1(config-pmap-c)# **set ip dscp** *dscp-markings*
    - R1(config-pmap-c)# **priority percent** *percentage*
    - R1(config-pmap-c)# **bandwidth percent** *percentage*

3. Apply the policy on interface
    - R1(config)# **interface** interface
    - R1(config-if)# **service-policy {output | input}** *pmap-name*

Security Fundamentals

Why security?

- Principles of CIA Triad form the foundation of security
    - Confidentiality
    - Integrity
    - Availability

Terms

- Vulnerability: Potential weakness
- Exploit: used to exploit vulnerability
- Threat: potential of vulnerability to be exploited
- Mitigation technique: things that protect against threats

Common attacks

- **DoS/ DDoS** (Distributed Denial of Service)
    - Target the availability of a system so users can't access it
    - Botnet - PCs affected by attacker in DDoS
    - E.g. TCP SYN flood
- **Spoofing**
    - Involve using fraudulent source IP/MAC addresses
    - E.g. DHCP Exhaustion
- **Reflection/amplification attacks**
    - involve spoofing a source IP address to cause a reflector to send lots of traffic to the target
- **Man-in-the-middle attacks**
    - an attacker intercepts traffic between the source and destination to eavesdrop and/or modify the traffic
    - ARP Poisoning
- **Reconnaissance attacks**
    - used to gather information on the target to perform future attacks
- **Malware**
    - malicious software such as viruses, worms, and trojan horses that infect a system
    - E.g. Virus (infect software), worms (don't need software), trojan horse
- **Social engineering attacks**
    - attacks that use psychological manipulation to target people and make them reveal info or perform an action
    - E.g. Spear Phishing (more targeted), Vishing (Voice phishing), Smishing (SMS Phishing)
    - E.g. Watering hole attacks (compromise commonly visited sites), tailgating
- **Password-related attacks**
    - attacks such as dictionary attacks and brute force attacks, used to guess the target's password
    - Types of attack
        - Guessing
        - Dictionary attack
        - Brute force

        - Strong password
            - >= 8 characters
            - Mix of upper/lower case
            - Mix of letters/numbers
            - >= 1 special character
            - Changed regularly

Multi-Factor Authentication (MFA)

- Use multiple of following
    - Something you know - password
    - Something you have - access card/phone notification
    - Something you are - biometrics

Digital Certificates

- Used for website to verify it is legitimate

Controlling & Monitoring users with AAA

- AAA (triple-A):
    - Authentication: verify user
    - Authorization: grant permissions
    - Accounting: record activities
- ISE (Identity Services Engine) Cisco's AAA server
- Usually use AAA server to provide AAA service
    - Servers usually support these 2 protocols
        - RADIUS
            - Open standard
            - UDP port 1812, 1813
        - TACACS+
            - Cisco proprietary
            - TCP port 49

User Program Element

- User awareness program
    - Make employees aware
    - E.g. fake phishing email
- User training
    - More formal than user awareness program
- Physical access control
    - Only allow authorized users to protected areas
    - Protects equipment from attackers

Switch Security

Port Security

Port Security

- Control which MAC address can enter switchport
- Can control
    - Number of devices
    - Specific MAC addresses
- By default, when enabled
    - 1 device allowed
    - MAC address dynamically learned
    - 'err-disabled'
- Need to be in static/trunk, cannot dynamic desirable/auto

Why port security?

- Control who and how many
- Normally to control number of devices
- Protect against threats like DHCP starvation attack

Violation Modes

- When unauthorized frame enter switchport
- There are 3
    - Shutdown
        - Interface shutdown, 'err-disabled'
        - Syslog/SNMP message
        - Violation counter = 1
    - Restrict
        - Interface not disabled, drop unauthorized traffic
        - Syslog/SNMP message
        - Violation counter increment by 1
    - Protect
        - Interface not disabled, drop unauthorized traffic
        - NO Syslog/SNMP message
        - NO change in violation counter
- NOTE
    - First step to re-enable is to remove unauthorized device

Secure MAC Address Aging

- Secure MAC address
    - MAC address learned dynamically/statically configured on port-security enabled interface
- By default, will not age
- Aging type

    - Absolute (default)
        - After timer ends, will remove address, even if still receiving
    - Inactivity
        - Timer restarts every time receives from address
- Secure Static MAC aging disabled by default

Sticky Secure MAC address

- Dynamic addresses will never age out

Config

- SW1(config-if)# **switchport mode** {**access** | **trunk**}
- SW1(config-if)# **switchport port-security**
- SW1(config-if)# **switchport port-security mac-address** *x.x.x.x*
- SW1(config-if)# **switchport port-security violation** {**restrict** | **protect**}

Aging

- SW1(config-if)# **switchport port-security aging time** *minutes*
- SW1(config-if)# **switchport port-security aging type** *aging-type*
- SW1(config-if)# **switchport port-security aging static**

Stick Secure

- SW1(config-if)# **switchport port-security mac-address sticky**

Show

- SW1# **show mac address-table secure**
- SW1# **show port-security**
- SW1# **show port-security interface** *interface*
- SW1# **show errdisable recovery**

ErrDisable Recovery

- SW1(config-if)# **errdisable recovery cause psecure-violation**
- SW1(config-if)# **errdisable recovery interval** *minutes*

DHCP Snooping

Intro

- Filter DHCP messages ONLY on untrusted ports
- Default: All ports untrusted

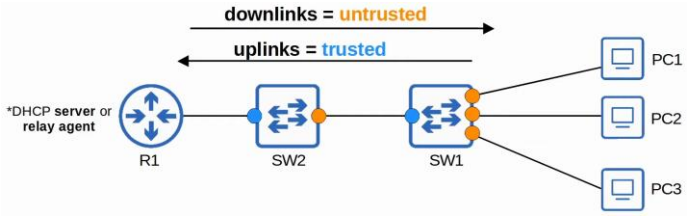
Protects against

- **DHCP starvation**
  - Attacker uses spoofed CHADDR (Client Hardware Address) and sends lots of DHCP Discover messages
  - Causes DHCP server's pool to be full, resulting in DoS to other devices
- **DHCP Poisoning (Man-in-the-middle)**
  - Attacker replies (Offer message) to client's Discover message (broadcast)
  - Attacker become default gateway of client
  - Attacker can examine/modify message before sending to actual default gateway

DHCP messages

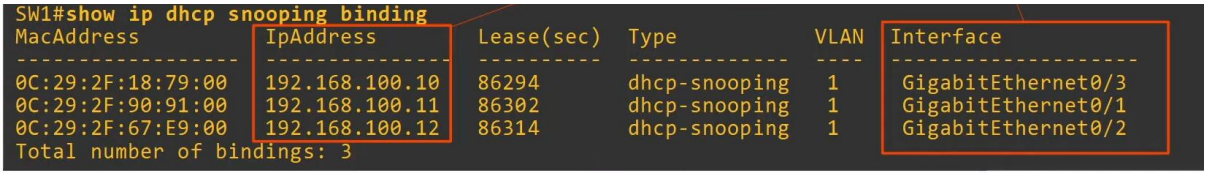
- **Server**
  - OFFER
  - ACK
  - NAK (decline client REQUEST)
- **Client**
  - DISCOVER
  - REQUEST
  - RELEASE (release server's IP address)
  - DECLINE (decline server's IP address)

DHCP Snooping Operations

- 

- If DHCP message received
  - **Trusted ports**
    - Forward w/o inspection
  - **Untrusted ports**
    - If from server, discard
    - If from client
      - DISCOVER/REQUEST
        - Check frame's source MAC address = DHCP message's CHADDR field
      - RELEASE/DECLINE
        - Check packet's source IP and receiving interface = DHCP Snooping Binding Table

DHCP Snooping Binding Table



- When a client leases an IP address, create new entry in binding table

Rate-Limiting

- If higher than set limit, interface err-disabled
- Manual reset: "shutdown, no shutdown"
- Auto reset: errdisable recovery
- Protect against DHCP exhaustion attacks

Option 82 (Information Option)

- Provide additional info regarding DHCP relay agents
- Relay agents can add it to DHCP messages being forwarded to DHCP server
- When DHCP snooping enabled, Cisco switches add Option 82 by default, even if not DHCP relay agent
- Default: switches will drop Option 82 messages on untrusted ports

Config

- SW1(config)# **ip dhcp snooping**
- SW1(config)# **ip dhcp snooping vlan** *vlan-number*
- SW1(config)# **no ip dhcp snooping information option**

Set limit

- SW1(config-if)# **ip dhcp snooping limit rate** *packets-per-second*
- SW1(config)# **errdisable recovery cause dhcp-rate-limit**

Set trusted port

- SW1(config-if)# **ip dhcp snooping trust**

Show

- SW1# **show ip dhcp snooping binding**

Dynamic ARP Inspection

Intro

- Very similar to DHCP Snooping, but check ARP messages instead
- Used to filter ARP messages received on untrusted port
- Only ARP messages checked
- Default: all ports untrusted
    - Ports connected to network device should be trusted

Prevents

- ARP Poisoning (Man-in-the-middle)

Operations

- Check ARP messages received on untrusted ports
    - Sender MAC = DHCP snooping binding table
    - Sender IP = DHCP snooping binding table
- ARP ACLs can be used for MAC addresses not in DHCP
- Can perform more checks
- Support rate-limiting
    - Shutdown if over limit
    - Prevent CPU from being overloaded

Optional Check

- "dst-mac"
    - Destination MAC: Ethernet Header = ARP body
- "src-mac"
    - Source MAC: Ethernet Header = ARP body
- "ip"
    - Check for invalid IP address
    - E.g. all 0, broadcast address, all multicast address
    - ARP request = check sender IP
    - ARP Reply = check sender/destination IP

Config

- SW1(config)# **ip arp inspection vlan** *vlan-number*
- SW1(config)# **ip arp inspection validate** (**src-mac** | **dst-mac** | **ip**)
- SW1(config-if)# **ip arp inspection trust**
- SW1(config-if)# **ip arp inspection limit rate** *packets* [**burst interval** *seconds*]
- SW1(config)# **errdisable recovery cause arp-inspection**

ACL

- SW1(config)# **arp access-list** *name*
- SW1(config-arp-nacl)# **permit ip host ip-address mac** *host mac-address*
- SW1(config)# **ip arp inspection filter** *arp-acl-name* **vlan** *vlan-number*

Show

- SW1# **show ip arp inspection**
- SW1# **show ip arp inspection interfaces**

Network Architecture

LAN

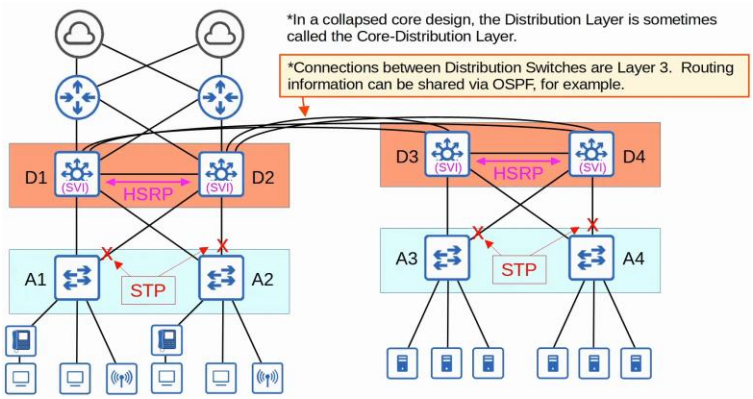
Common Terms

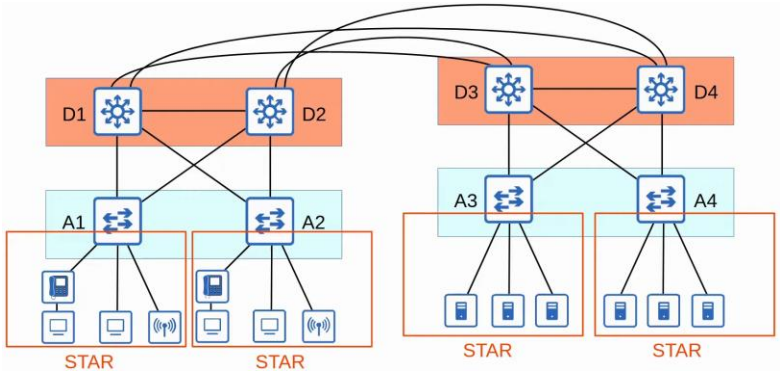
- Star
- Full/Partial Mesh

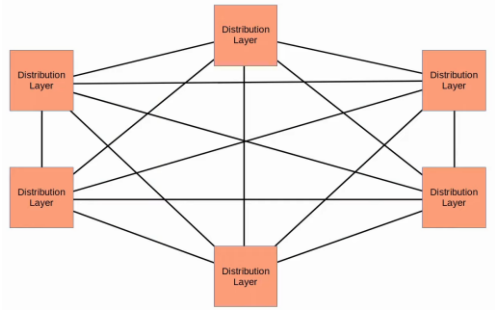
Campus LAN Design

- 2-Tier ("Collapsed Core")
    - Access Layer
    - Distribution Layer
- 3-Tier
    - Access Layer
    - Distribution Layer
    - Core Layer
- Use 3-tier when there are many distribution layers
    - They are full mesh topology, so need lots of connection when many distribution layer
    - Cisco recommend if 3 distribution layer in 1 location
- **Access Layer**
    - To end hosts
    - Many ports
    - QoS markings
    - Security services
    - PoE
- **Distribution Layer (Aggregation Layer)**
    - Aggregate connection from Access Layer switches
    - Typically, border btw Layer 2 & 3
    - Connects to services such as Internet (*for tier 2)
- **Core Layer**
    - Connect Distribution Layer together
    - Focus on speed
    - Avoid CPU intensive operations
    - All Layer 3 connections
    - Maintain connectivity throughout LAN even if device fails

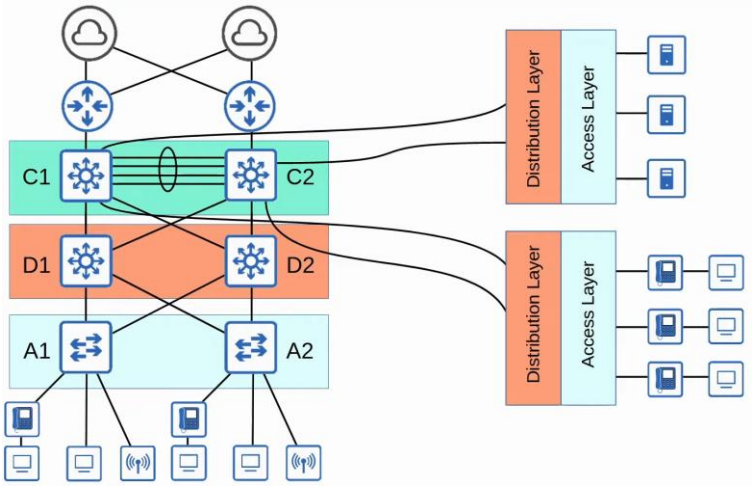
2-Tier

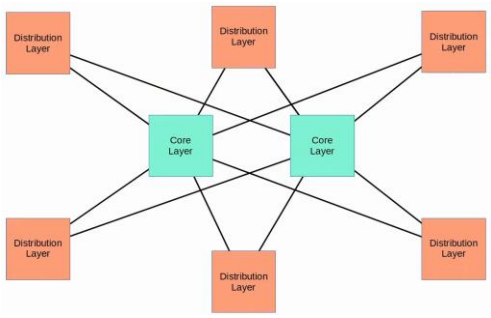

*In a collapsed core design, the Distribution Layer is sometimes called the Core-Distribution Layer.
*Connections between Distribution Switches are Layer 3. Routing information can be shared via OSPF, for example.
D1
(SVI)
HSRP
(SVI)
D2
D3
(SVI)
HSRP
(SVI)
D4
A1
STP
A2
A3
STP
A4


D1
D2
D3
D4
A1
A2
A3
A4
STAR
STAR
STAR
STAR


Distribution Layer
Distribution Layer
Distribution Layer
Distribution Layer
Distribution Layer
Distribution Layer

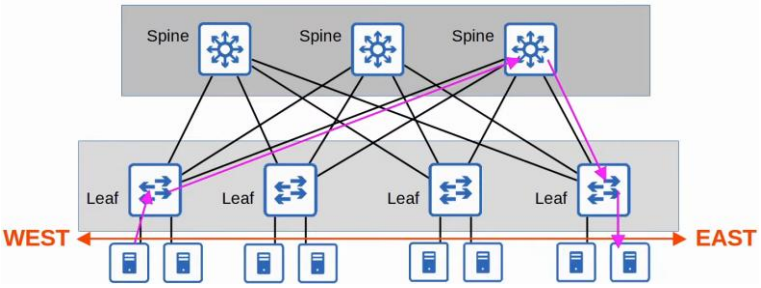
3-Tier


C1
C2
D1
D2
A1
A2
Distribution Layer
Access Layer
Distribution Layer
Access Layer



Spine-Leaf Architecture

- Used in data centres
- With virtual servers, applications deployed in a distributed manner (across multiple server)
- Rules
    - Every leaf connect to every spine
    - Every spine connect to every leaf
    - Leaf/spine don't connect to other leaf/spine
    - End host only connect to leaf
- Path taken is random
- Hop count is same, except for servers connected to the same leaf
- Easy to scale



Small Office / Home Office (SOHO) Networks

- All network functions in 1 device (home/wireless router)
- Device serve as
    - Router
    - Switch
    - Firewall
    - Wireless Access Point
    - Modem

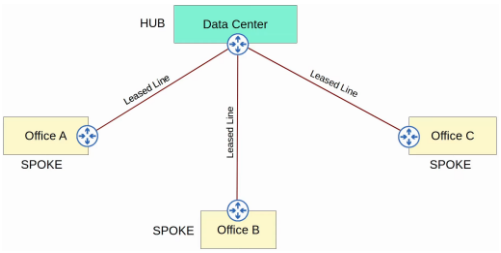
WAN

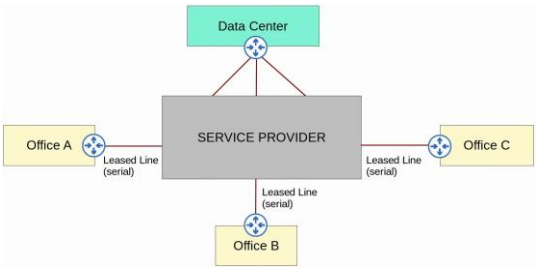
Wide Area Network (WAN) Architecture

- Large geographic area
- Connect geographically separated LANs
- VPNs can be used to create private WAN connections over public/shared networks

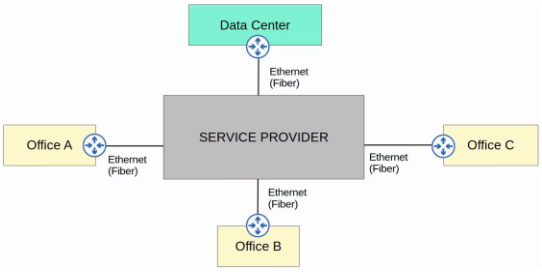
WAN over dedicated connection (Leased Line)

- Hub and spoke topology

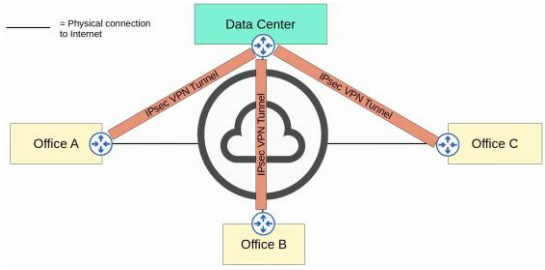




WAN connection via Ethernet (Fiber)



WAN over shared infrastructure (Internet VPN)



Leased Line

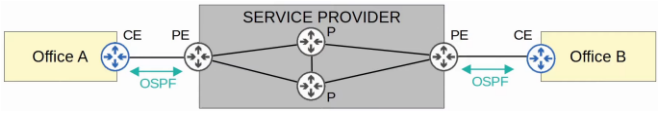
- Dedicated physical link
- Use serial connection (PPP, HDLC)
- Ethernet more popular now
- T1: 1, T2: 6, T3: 44
- E1: 2, E2: 8, E3: 34

Multi-Protocol Label Switching (MPLS)

- Use labels on packets to create VPNs over MLPS infrastructure
- Terms
    - Customer Edge (CE) Router
    - Provider Edge (PE) Router
    - Provider core (P) router
- Labels added by PE when receive by CE
    - Used to route within service provider network ONLY

Layer 3 MPLS

- CE router don't use MPLS, only PE and P
- CE and PE routers peer using OSPF, share routing info
    - CE can use PE as next hop for static routing
- A will have route to B, vice versa

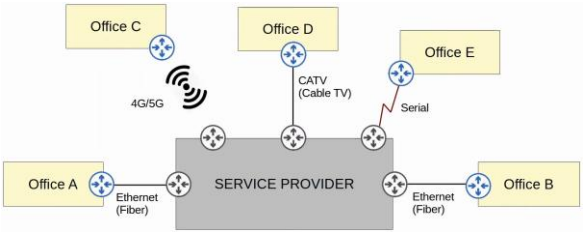


Layer 2 MPLS

- Service provider network transparent to CE
    - Acts like a switch
- CE WAN interfaces in same subnet
- Will directly peer each other if routing protocol used

Connections

- Different technologies can be used to connect to service provider's MLPS network



Internet Connections

- Many ways to connect to Internet
    - Private WAN technologies (Leased line, MPLS VPNs)
    - CATV, DSL
    - Fiber Optic Ethernet (Popular now)

Digital Subscriber Line (DSL)

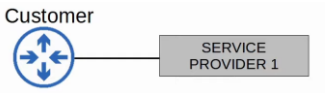
- Internet through phone line
- DSL Modem (modulator-demodulator) needed
    - Convert data to suitable format
    - Can be separated/built into home router

Cable Television (CATV)

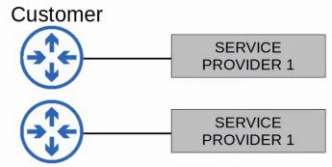
- Internet via CATV lines
- Need modem, like DSL

Redundant Internet Connections

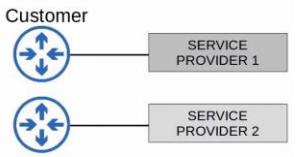
- **Single homed**



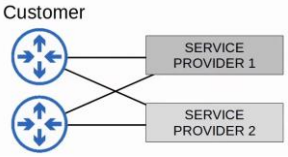
- **Dual homed**



- **Multi homed**



- **Dual Multi-homed**

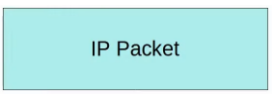


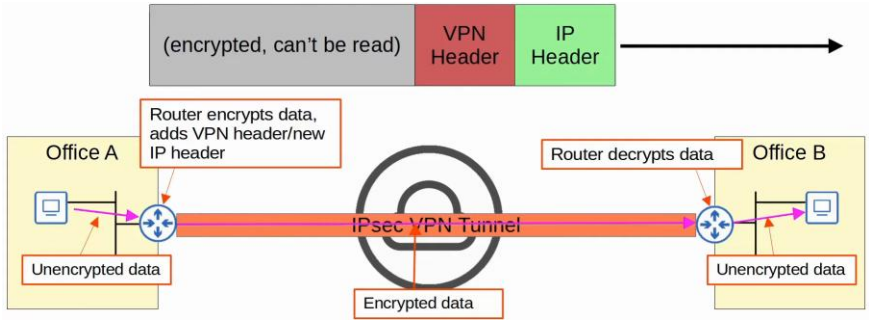
Internet VPN

- Private WAN (leased line, MPLS) are secure
    - Dedicated physical connection (leased line)
    - MPLS tag
- Internet not safe, no built-in security feature
- Use VPN, 2 types
    - Site-to-Site VPNs using IPSec
    - Remote-accessing VPNs using TLS

Site-to-Site VPNs (IPSec)

- It is a VPN btw 2 devices and used to connect them over Internet
- VPN 'tunnel' created btw the 2 devices
    - Encapsulates original packet with VPN header and new IP header
    - When using IPSec, original packet encrypted

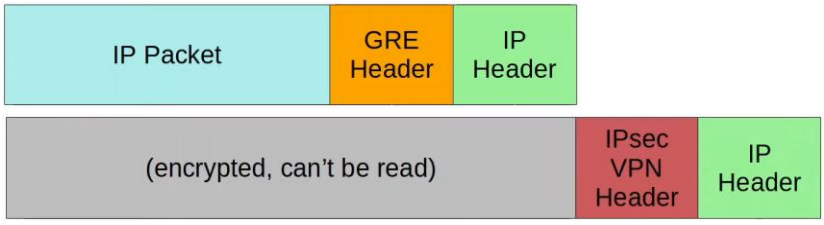




- **How it works**
  - Sender combine original packet and session key (encryption key) and encrypt them
  - Sender encapsulate encrypted packet with VPN and new IP header
  - Sender send to other device
  - Receiver decrypts and forward original packet to destination
- For S-t-S, tunnel is only formed btw 2 endpoints (e.g. btw 2 routers in diagram)
  - Other devices in site (office) don't need create a VPN for themselves
  - They send unencrypted data to router
- **Limitations**
  - IPSec don't support broadcast/multicast traffic, only unicast
    - Cannot use routing protocols (e.g. OSPF) over tunnels
    - Solution: "GRE over IPSec"
  - Configuring a full mesh of tunnels btw many sites labour-intensive
    - Solution: Cisco's DMVPN

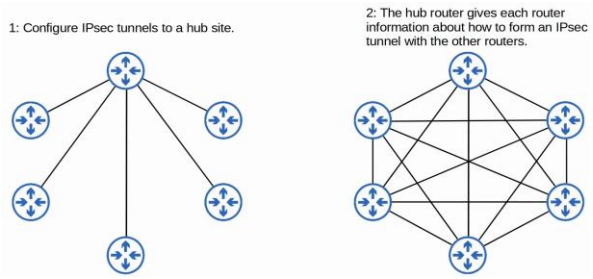
GRE Over IPSec

- Generic Routing Encapsulation (GRE)
  - Create tunnel like IPSec
  - Bad: no encryption
  - Good: able to encapsulate wide variety of Layer 3 messages and broadcast/multicast messages
- GRE over IPSec: encryption for GRE
  - Original packet encapsulated with GRE and new IP header
  - GRE packet encrypted
  - Encapsulated with IPSec VPN and new IP header



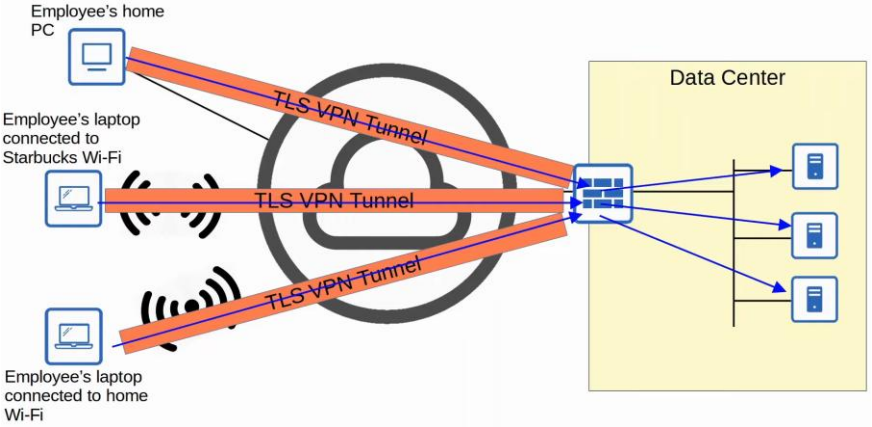
DMVPN

- Dynamic Multipoint VPN
  - Allow routers to create full mesh of IPSec tunnels dynamically w/o manual config
  - Config simplicity of hub-and-spoke (1 spoke = 1 tunnel)
  - Efficiency of direct spoke-to-spoke communication (spoke -> spoke w/o going through hub)



Remote-Access VPNs

- Allow end device access company's internal resource over Internet
- Normally use TLS (Transport Layer Security)
- How it works
    - VPN client software installed on end devices
    - End device form secure tunnel to 1 of company's router/firewall acting as TLS server
    - Can access resource w/o being directly connected to network
- Also encrypt data like in IPSec



Differences

- Site-to-Site VPNs
    - Use IPSec
    - Provide service to many devices within the sites they are connecting
    - Permanent
- Remote Access VPNs
    - Use TLS
    - Provide service to the 1 end device VPN client software is installed on
    - On-demand

Virtualization

Virtualization

Before Virtualization

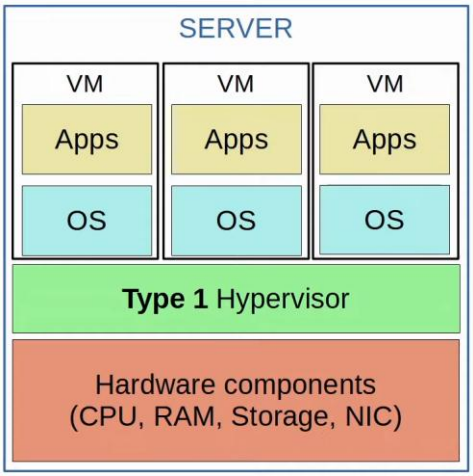
- 1 server = 1 OS
- Inefficient
    - Expensive
    - Resource under utilized

With Virtualization

- 1 server = more than 1 OS
- 1 OS instance -> Virtual Machine
- Hypervisor
    - Aka VMM (Virtual Machine Monitor)
    - Manage & allocate hardware to VMs

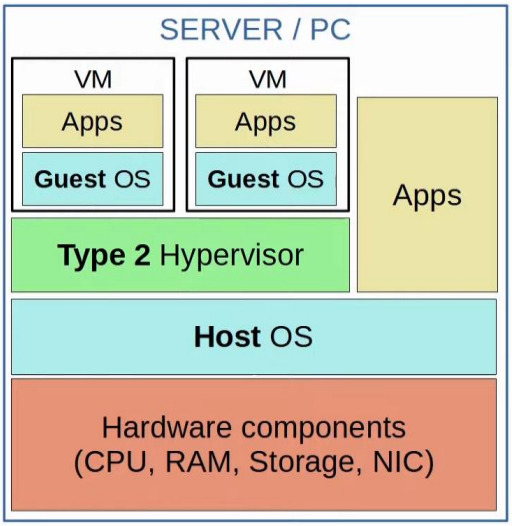
Type 1 hypervisor

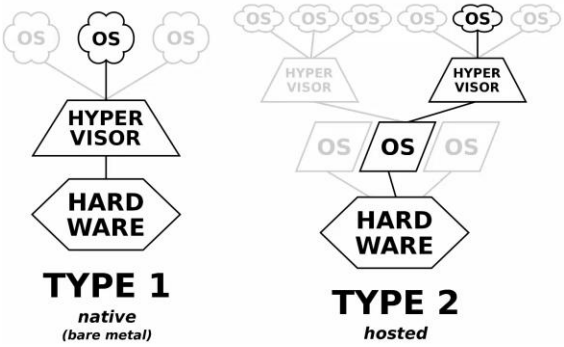
- Aka bare-metal/native hypervisors
- Hypervisor run directly above hardware



Type 2 Hypervisor

- Aka Hosted hypervisor
- Hypervisor run like a normal CPU program
- Usually used on PCs



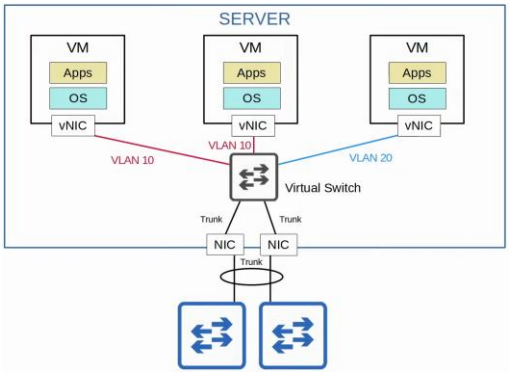


Benefits

- Partitioning: 1 machine = multiple OS
- Isolation: 1 VM fail won't affect others
- Encapsulation: Save and copy VMs like files
- Hardware independence: Provision/migrate any VM to any physical server

Connecting VMs to Networks

- VMs connected to each other via virtual switch on hypervisor
    - Can operate trunk/access ports and separate VLANs
    - vSwitch connect to physical NIC



Cloud Services

- Traditional IT infrastructure use
    - On-Premise
    - Colocation
- Main features
    - **5 essential characteristics**
        - On-demand self-service
        - Broad network access
        - Resource pooling
        - Rapid elasticity
        - Measured service
    - **3 service models**
        - Software as a Service (SaaS)
        - Platform as a Service (PaaS)
        - Infrastructure as a Service (IaaS)
    - **4 deployment models**
        - Private cloud
        - Community cloud
        - Public cloud
        - Hybrid cloud

Benefits

- Cost
- Global scale
- Speed/Agility
- Productivity
- Reliability

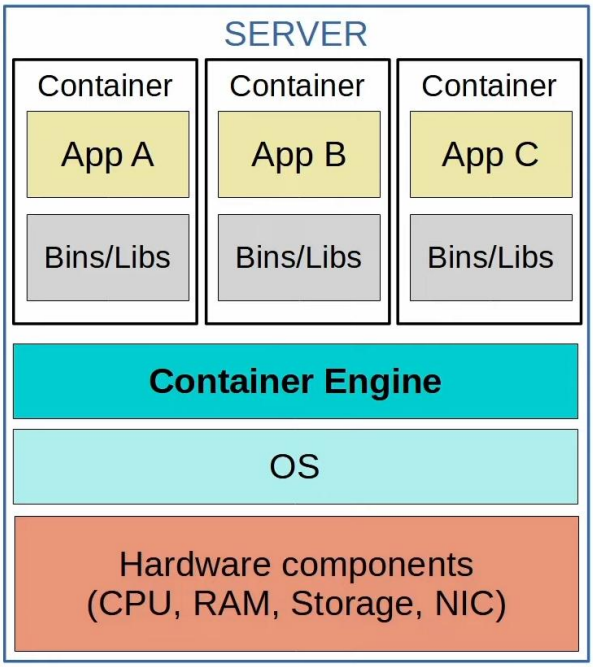
Connecting to cloud resources

- Via
    - Private WAN/Service Provider
    - Internet
    - Internet (IPSec VPN tunnel)

Containers

- Software packages that contain
    - An App
    - All dependencies
- Containers run on Container Engine (e.g. Docker Engine)
    - Engine runs on host OS (e.g. Linux)
- Container Orchestrator

- Software platform that automates deploying/managing containers
- E.g. Kubernetes, Docker Swarm



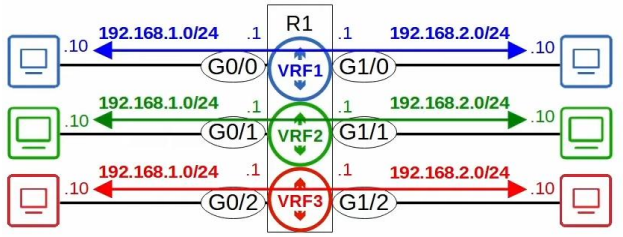
VM vs Container

- Boot up time
  - VM: take minutes as each VM run its own OS
  - Containers: in milliseconds
- Space
  - VM: use more disk space (GB)
  - Container: very little disk space (MB)
- Resources
  - VM: use more CPU/RAM resources (each VM run its own OS)
  - Container: use fewer CPU/RAM (shared OS)
- Portability
  - VM: portable and can move btw physical systems running the same hypervisor
  - Container: more portable, they are smaller, faster to boot up, and Docker containers can be run on nearly any container service
- Reliability
  - VM: more isolated as each VM runs its own OS
  - Container: less isolated because they all run on the same OS; if the OS crash, all containers running on it are affected

VRF (Virtual Routing & Forwarding)

- Divide 1 router into multiple virtual router
- Build separate routing tables
- This kind is VRF-lite (w/o MPS)
- Used by service providers to allow 1 device to carry traffic from multiple customers

- Each customer's traffic is isolated from the others
- Customer IP addresses can overlap w/o issues



Wireless

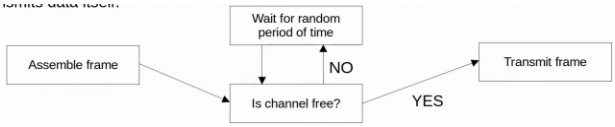
Intro

Wireless Networks

- IEEE 802.11

Issues with wireless networks

1. Devices will receive all frames
    - Privacy/security issue
    - CSMA/CA used to facilitate half-duplex communication
        - A device will wait for other devices to stop transmitting before it transmits data itself



1. They are regulated
2. Wireless signal coverage must be considered
    - Signal
        - Range
        - Absorption
        - Reflection
        - Refraction
        - Diffraction
        - Scattering
3. Other devices on the same channel can cause interference

Radio Frequency (RF)

- Apply AC to antenna to create Wi-Fi signals
- Radio Freq: 30 Hz - 300 GHz

Radio Frequency Bands

- Wi-Fi uses 2 main bands (frequency ranges)
    - 2.4GHz band
        - 2.400GHz - 2.4835GHz
    - 5GHz band
        - 5.150GHz - 5.825GHz
- 2.4 GHz better range and penetration of obstacles
- Wi-Fi = 6 GHz

Channels

- Each band divided into smaller channels
- For 2.4 GHz, use channels 1, 6 and 11

802.11 Standards

Standard	Frequencies	Max Data Rate (theoretical)	Alternate Name
802.11	2.4 GHz	2 Mbps	
802.11b	2.4 GHz	11 Mbps	
802.11a	5 GHz	54 Mbps	
802.11g	2.4 GHz	54 Mbps	
802.11n	2.4 / 5 GHz	600 Mbps	'Wi-Fi 4'
802.11ac	5 GHz	6.93 Gbps	'Wi-Fi 5'
802.11ax	2.4 / 5 / 6 GHz	4*802.11ac	Wi-Fi 6'

Service Sets

- Service sets: groups of wireless network devices
- 3 main types
    - Independent
        - IBSS
    - Infrastructure
        - BSS
        - ESS
    - Mesh
        - MBSS
- Service Set Identifier (SSID)
    - Wi-Fi name
    - All device in 1 service set = same SSID
    - Human readable
    - Not unique

Independent Basic Service Set (IBSS)

- Wireless network where devices connect directly w/o using an AP

- AKA 'Ad hoc network'
- E.g. AirDrop

Basic Service Set (BSS)

- Client connect to each other via AP
    - Cannot connect directly to each other
- BSSID used to identify AP
    - MAC address of AP
    - Other AP can have same SSID, not BSSID
- Client must associate with AP to be in BSS
- BSA (Basic Service Area) is the range of the signal of the AP
- BSS is the group of devices

Extended Service Set (ESS)

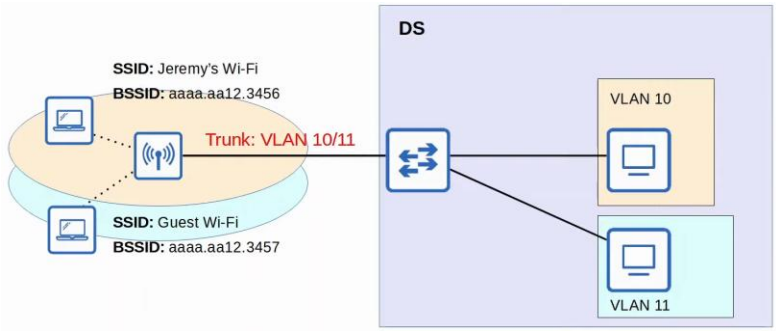
- Larger version of BSS
- APs of different BSS connected by a wired network
    - The BSS have
        - Same SSID
        - Unique BSSID
        - Different channel to avoid interference

Mesh Basic Service Set (MBSS)

- Mesh APs use 2 radios
    - 1 to provide BSS to client
    - 1 for 'backhaul network' to route traffic
- 2 types of AP
    - RAP (Root Access Point)
        - Connected to wired network
        - At least 1
    - MAP (Mesh Access Point)

Distribution System (DS)

- Wireless used to access wired network
- The wired part is called DS
- 1 BSS/ESS = 1 VLAN in wired network
- Possible, 1 AP = multiple BSS
    - Each WLAN mapped to separate VLAN
    - Each WLAN have unique BSSID, +1



Additional AP Operation Modes

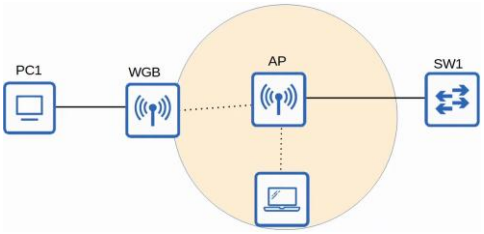
- The AP modes are
    - Repeater
    - Workgroup Bridge (WGB)
    - Outdoor Bridge

Repeater

- Extend range
- Use 2 radio of different channel to reduce interference
    - 1 for receiving
    - 1 for transmitting

Workgroup Bridge (WGB)

- Provide wireless connectivity to wired clients
- 2 types
    - Universal WGB (uWGB)
        - 802.11 standard
        - 1 client
    - WGB
        - Cisco proprietary
        - >1 client

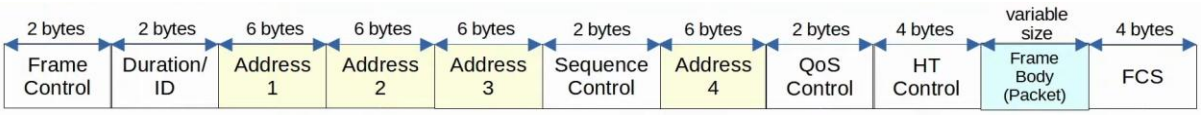


Outdoor Bridge

- Wireless connection over long distance
- Can be point-to-point or point-to-multipoint

Wireless Architecture

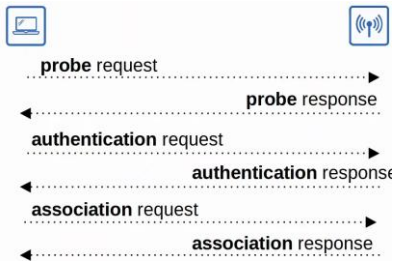
802.11 Frame Format



- **Frame control**
  - Info such as message/sub type
- **Duration/ID**
  - Depend on message type
    - Time dedicated for transmission (micro s)
    - ID for the association/connection
- **Addresses**
  - 4 types
    - DA (Destination Address): Original
    - SA (Source Address): Original
    - RA (Receiver Address): interim, hop-to-hop
    - TA (Transmitter Address): interim
- **HT (High Throughput) Control**
  - 802.11n - HT Wi-Fi
  - 802.11ac - Very HT (VHT) Wi-Fi

802.11 Association Process

- Device must be associated with AP before sending data
- 3 connection states
  - Not authenticated, associated
  - Authenticated, not associated
  - Authenticated, associated (can send traffic)
- 2 ways for device to scan for BSS
  - **Active scanning**
    - Device send Probe Request
    - Listen for Probe Response from AP
  - **Passive scanning**
    - Device listen for Beacon message from AP
    - Beacon message send periodically by AP to advertise BSS



802.11 Message Types

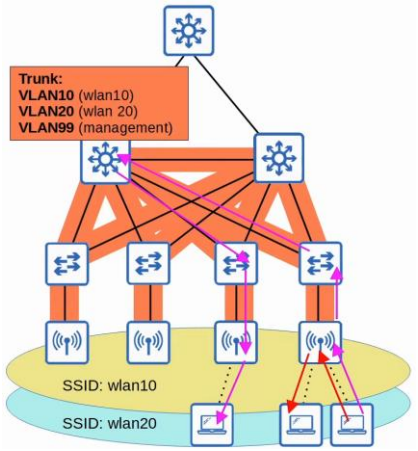
- There are 3
    - **Management**
        - Used to manage the BSS
        - E.g. Beacon, Probe request/response, Authentication, Association
    - **Control**
        - Used to control access to a medium (radio frequency)
        - Assist with delivery of management and data frames
        - E.g. RTS (Request to Send), CTS (Clear to Send), ACK
    - **Data**
        - Used to send actual data packets

Wireless AP Deployment

- There are 3 main wireless AP deployment methods
    - Autonomous
    - Lightweight
    - Cloud-based

Autonomous APs

- Self-contained systems that don't rely on Wireless LAN Controller (WLC)
- Configured individually & manually
- No management APs
- Connect to wired network using trunk link
    - Separate VLAN for management and data
- Each VLAN spread through entire network
    - Bad
- APs can function as repeater, WBG, outdoor bridge

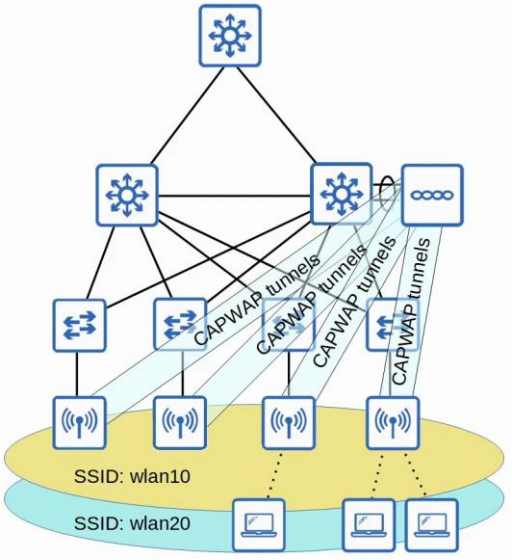


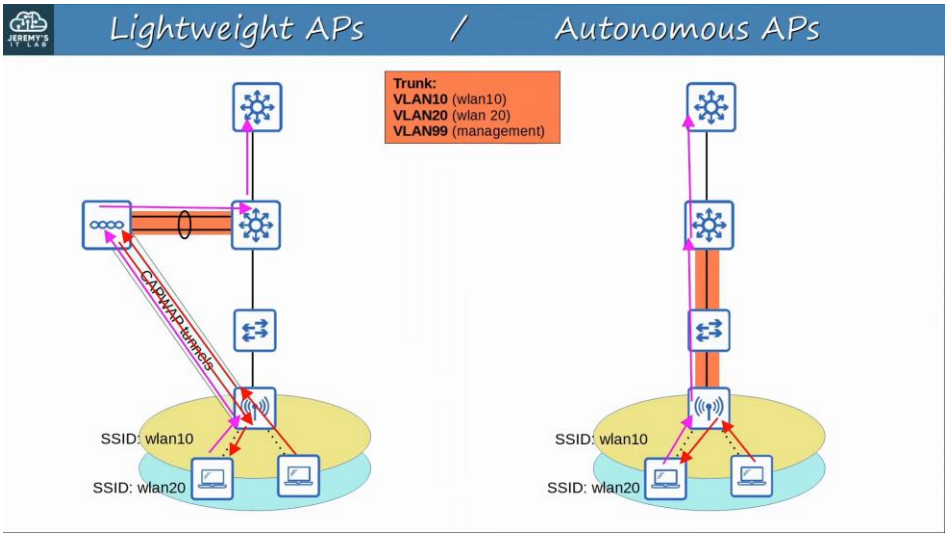
xx

Lightweight APs

- 2 devices used
    - **Lightweight APs**
        - Handle real-time operations
    - **WLC**

        - Carry out other functions
        - Centrally config APs
        - Can be located in same/different subnet/VLAN as APs
- "Split-MAC Architecture"
- WLC and APs use digital certificate to authenticate each other
- CAPWAP (Control and Provisioning of Wireless Access Points)
    - Used to communicate
    - 2 tunnels created
        - **Control tunnel**
            - UDP port 5246
            - Management traffic
            - Encrypted
        - **Data tunnel**
            - UDP port 5247
            - Data traffic
            - Sent straight to WLC through tunnel, won't go to wired network
            - Not encrypted, can encrypt with DTLS (Datagram Transport Layer Security)
- Switch ports, not trunk

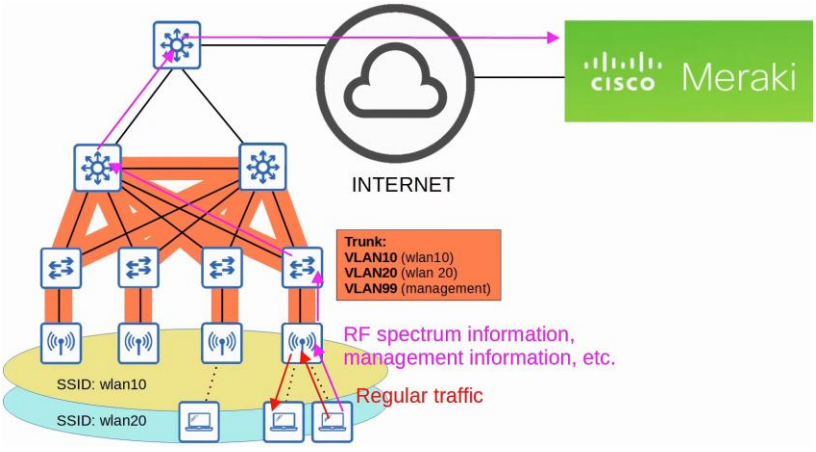




- **Lightweight APs Operation Modes**
  - Local
  - FlexConnect: can switch to wired if WLC down
  - Sniffer: capture for wireshark
  - Monitor: capture for checking
  - Rogue detector: check wired
  - SE-Connect (Spectrum Expert Connect): check signals
  - Bridge/Mesh: long distance, mesh
  - Flex plus Bridge

Cloud-based APs

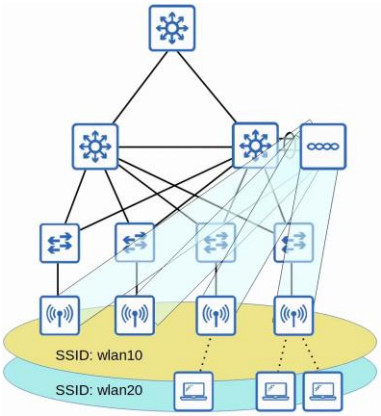
- Basically autonomous APs managed in cloud
  - Data traffic within wired network
  - Management traffic goes to cloud
- E.g. Cisco Meraki

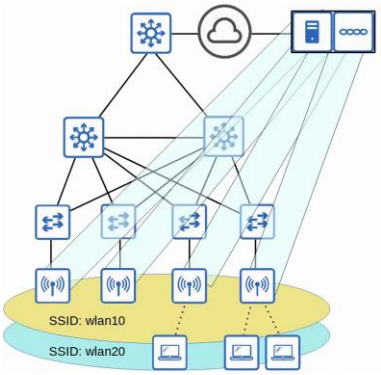


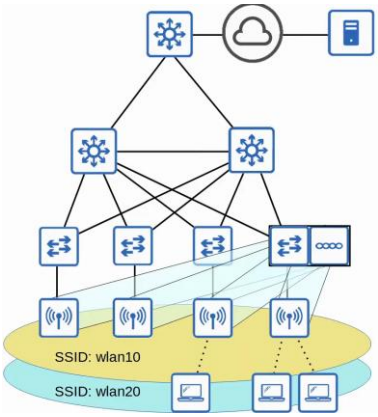
Deployment Methods

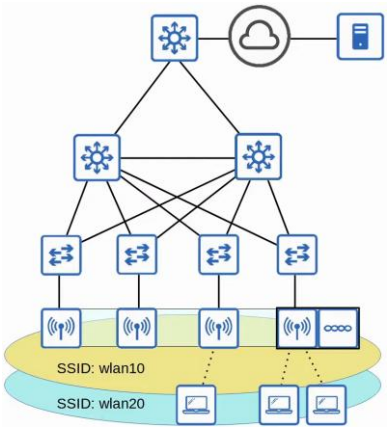
- For "Split-MAC Architecture" (Lightweight APs)

- 4 methods
    - **Unified**
        - Hardware appliance for WLC, ~6000 APs
    - **Cloud-based**
        - WLC is a VM running on a server, ~3000 APs
    - **Embedded**
        - Embedded within a switch, ~200 APs
    - **Mobility Express**
        - Embedded within AP, ~ 100 APs









Wireless Security

Intro

- Everyone within range can receive wireless traffic
    - Important to secure the traffic
- 3 main concepts
    - **Authentication**
        - Client must be authenticated before associating with AP
        - Client should authenticate AP, prevent from malicious AP
        - E.g. Password, Certificate, etc
    - **Encryption**
        - All wireless traffic
        - All device in WLAN use same protocol
            - Each client - unique key
            - Group key - AP and all clients have, for AP to send to all clients
    - **Integrity**
        - Ensure message remains unmodified in transit
        - MIC (Message Integrity Check) used
        - MIC added to message -> message encrypted -> receive and decrypt -> use message to calculate MIC -> check calculated MIC with MIC in message

Authentication Methods

1. Open Authentication
    - AP accepts all authentication requests
    - May have additional authentication after
    - E.g. Airport WiFi, Starbucks WiFi

1. WEP (Wired Equivalent Privacy)
    - Authentication & Encryption
    - Encryption
        - RC4 algorithm

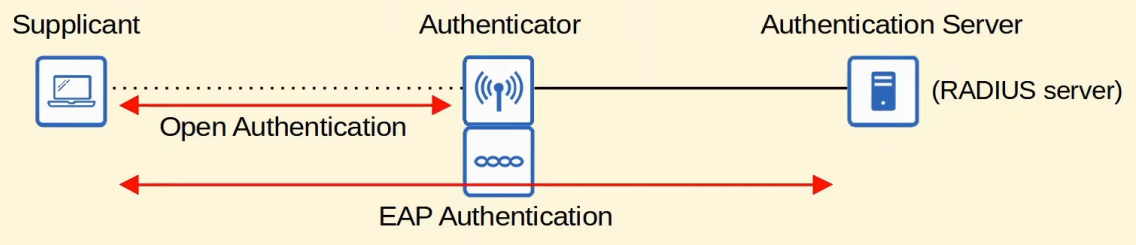
        - 'Shared-key', AP and client same key
        - Easily cracked
    - Authentication
        - AP send 'challenge phrase' -> client decrypt using key, send decrypted msg -> AP compare the message with original

EAP (Extensible Authentication Protocol)

- Authentication framework
- Integrated with 802.1X

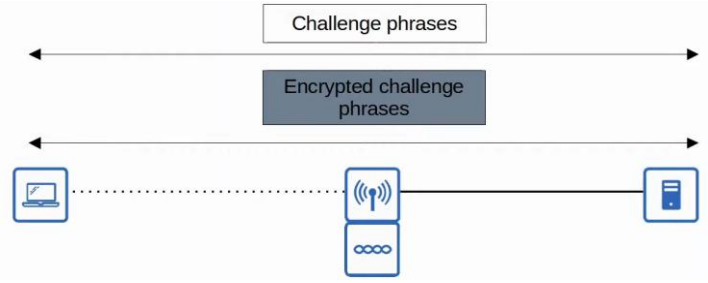
802.1X

- Limit network access to clients until authenticated
- 3 entities
    - **Supplicant**: Client
    - **Authenticator**: AP
    - **Authentication Server**: Receive client credentials and deny/permit access
- Can only authenticate when connected through open authentication



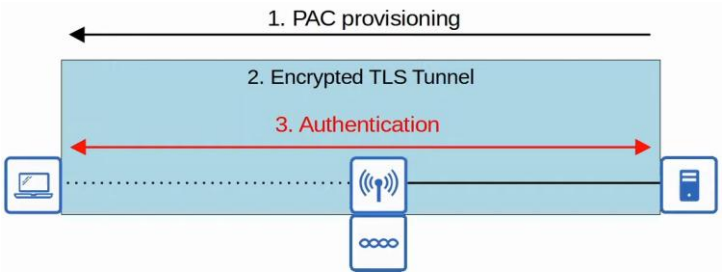
LEAP (Lightweight EAP)

- Client provide username and password
- Both client and server send challenge phrase
- Dynamic WEP keys
- Vulnerable, bad to use



EAP-FAST (EAP Flexible Authentication via Secure Tunnelling)

- 3 phases
    1. Server generate PAC (Protected Access Credential) and pass to client
    2. Secure TLS tunnel established btw client and server
    3. In TLS tunnel (encrypted), server can authenticate client

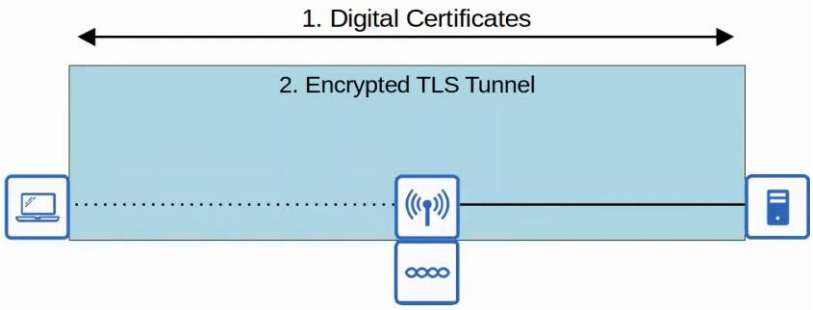


PEAP (Protected EAP)

- Same like EAP-FAST
    - But use digital certificate instead of PAC
- Server still authenticates client in tunnel, since only server generate certificate to client

EAP-TLS (EAP Transport Layer Security)

- Server and client have digital certificate
- Most secure, but difficult
- No authentication in tunnel
    - Authenticated with certificate already
- Tunnel used to exchange encryption key information



Encryption/Integrity Methods

- TKIP (Temporal Key Integrity Protocol)
- CCMP (Counter/CBC-MAC Protocol)
- GCMP (Galois/Counter Mode Protocol)

TKIP (Temporal Key Integrity Protocol)

- Temporary upgrade to WEP
- Added features
    - MIC, includes
        - Sender MAC
        - Timestamp
        - TKIP sequence number
    - Unique key for every frame
- WPA

CCMP (Counter/CBC-MAC Protocol)

- More secure than TKIP
- WPA2

- Must be supported by hardware
- 2 different algorithm for encryption and MIC
    - AES (Advanced Encryption Standard)
        - 'Counter' mode
        - Most secure encryption in the world
        - Multiple modes, CCMP uses 'counter'
    - CBC-MAC (Cipher Block Chaining Messages Authentication Code)
        - Used as MIC

Galois/Counter Mode Protocol (GCMP)

- More secure and efficient than CCMP
- Higher data throughput than CCMP
- WPA3
- 2 algorithms
    - AES counter mode
    - GMAC (Galois Message Authentication Code)
        - MIC

Wi-Fi Protected Access (WPA)

- 3 WPA certification
    - WPA
    - WPA2
    - WPA3
- 2 Authentication Modes
    - **Personal Mode**
        - PSK (Pre-Shared Key) for authentication
        - Common for small networks
        - 4-way handshake is used for authentication, and the PSK is used to generate the encryption keys
    - **Enterprise Mode**
        - 802.1X is used with an authentication server (RADIUS Server)
        - All EAP methods supported

WPA

- Protocols
    - TKIP - encryption/MIC
    - Enterprise/Personal mode

WPA2

- Protocols
    - CCMP - encryption/MIC
    - Enterprise/Personal mode

WPA3

- Protocols

- GCMP - encryption/MIC
- Enterprise/Personal mode
- Additional security features
    - PMF (Protected Management Frames)
        - Protect 802.11 frames from eavesdropping/forging
    - SAE (Simultaneous Authentication of Equals)
        - Protects the 4-way handshake in personal mode
    - Forward Secrecy
        - Prevent data from being decrypted after transmitted over air

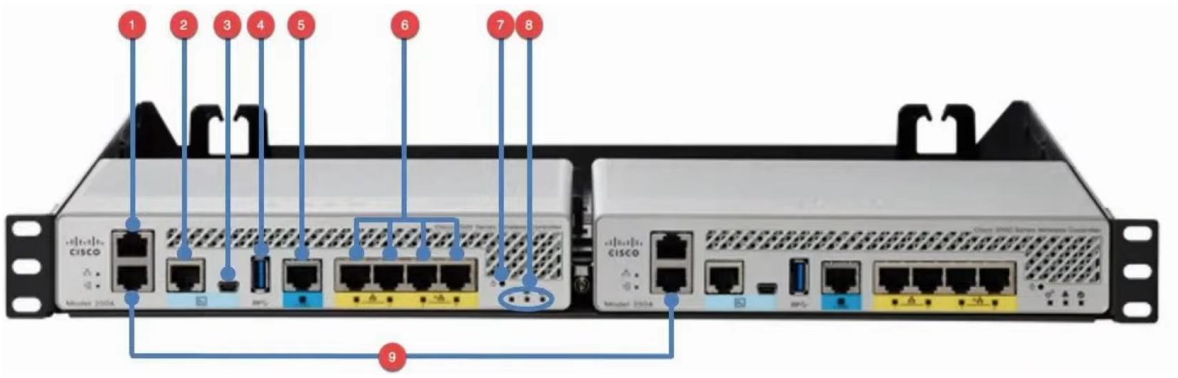
WLC

WLC Ports/Interface

- WLC Ports
    - Physical ports that cables connect to
- WLC Interface
    - Logical interfaces within the WLC (i.e. SVIs on a switch)

Type of Ports

- **Service Port**
    - Dedicated management port
    - Used for out-of-band management
    - Must connect to a switch access port because it only supports 1 VLAN
    - Can be used to connect to the device while it is booting, perform system recovery, etc
- **Distribution System Port**
    - Standard network ports that connect to the 'distribution system' (wired network) and are used for data traffic
    - Usually connect to trunk ports, and if multiple distribution ports are used, they can form a LAG
- **Console Port**
    - Standard console port
    - RJ45 / USB
- **Redundancy Port**
    - Used to connect to another WLC to form a high availability (HA) pair



1. Service Port
2. Console Port (RJ45)
3. Console Port (USB)
4. USB (for software updates)
5. Distribution system port (multi-gigabit)
6. Distribution system ports (1-gig)
7. Reset button
8. Status LED
9. Redundancy port

Types of Interfaces

- **Management interface**
    - Used for management traffic such as Telnet, SSH, HTTP, HTTPS, RADIUS authentication, NTP, Syslog, etc
    - Layer 2 LWAPP communication btw WLC and Aps
    - Communicate btw other WLC in network
- **Redundancy management interface**
    - When 2 WLCs are connected by their redundancy ports, 1 WLC is 'active' and 1 is 'standby'
    - This interface can be used to connect and manage the 'standby' WLC
- **Virtual interface**
    - Used when communicating with wireless clients to relay DHCP requests, perform client web authentication, etc
    - IP the same across WLCs, support seamless roaming
    - Support mobility management
- **Service port interface**
    - If the service port is used, this interface is bound to it and used for out-of-band management
    - Used for maintenance
- **AP-manager interface**
    - Control layer 3 communications btw WLC and AP
    - CAPWAP tunnels formed btw AP-manager interface and AP
    - Must be unique since communicate with AP on wireless network

- **Dynamic interface**
    - Used to map a WLAN to a VLAN
    - E.g. traffic from the 'internal' WLAN will be sent to the wired network from the WLC's 'internal' dynamic interface

QoS

- Platinum: voice
- Gold: video
- Silver: best effort
- Bronze: background

Network Automation

Network Automation

Intro

- Benefits
    - Human error reduced
    - Easier to scale networks
    - Network policy compliance
    - Improved efficiency

Logical Planes

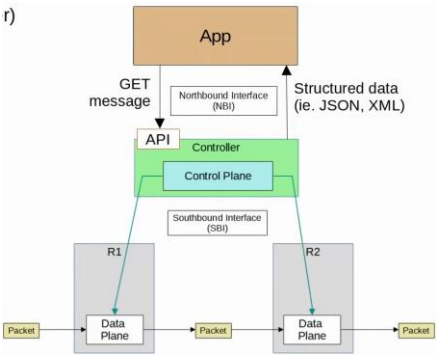
- Data Plane / Forwarding Plane
    - All tasks involved in forwarding traffic
    - E.g. NAT, ACL
- Control Plane
    - Controls what the data plane does
    - E.g. OSPF, STP, ARP
- Management Plane
    - Don't affect traffic forwarding
    - Consists of protocols that to manage device
        - SSH/Telnet
        - Syslog
        - SNMP
        - NTP

Hardware Management

- ASIC (Application-Specific Integrated Circuit) used for data plane operations
- Tables stored in TCAM (Ternary Content-Addressable Memory)
    - Tables also called CAM table
    - E.g. MAC address table
- If receive control/management traffic (destined for itself), use CPU
- If receive data traffic to be forwarded, use ASIC

SDN (Software-Defined Networking)

- Aka Controller-based networking
- Centralized control plane to a single controller
- Controller interact programmatically with network devices through API



Southbound Interface (SBI)

- Used for communication btw controller and network devices
- Consist of communication protocol and API
- Examples
    - OpenFlow
    - Cisco OpFlex
    - Cisco onePK
    - NETCONF

Northbound Interface (NBI)

- Allow user (us) to interact with controller
- REST API used on controller for apps to interact with

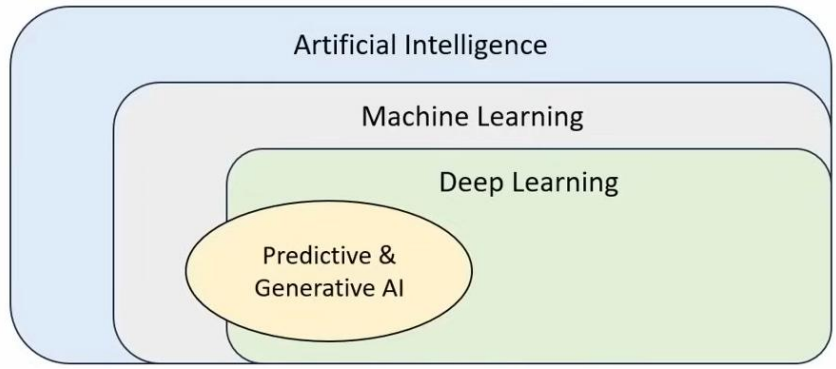
AI & Machine Learning

Intro

- AI: uses computer to simulate intelligence
- ML: focus on enabling computers to learn from data and improve w/o explicit programming

Types of ML

- **Supervised Learning**
    - Train on labelled set, predict new dataset
- **Unsupervised Learning**
    - Train on unlabelled set, group data based on similar features
- **Reinforcement Learning**
    - Reward/Penalize the agent's action in an environment
    - Used for games, e.g. chess
- **Deep learning**
    - Uses artificial neural network, data pass through several layers of nodes
    - Can be trained using the above learning methods



AI

- **Predictive AI**
  - Predict future given past datasets
  - Applications: traffic forecasting, security threat detection, predictive maintenance
- **Generative AI**
  - Use ML to learn from existing data to create new data
  - Application: network documentation, config generation, network design, troubleshooting, script generation

Cisco Catalyst Center

- Features include
  - AI Network Analytics
    - Analyse network
  - Machine Reasoning Engine
    - Find issues and give solution
  - AI Endpoint Analytics
    - Devices analysis (no. of devices, etc)
  - AI-enhanced Radio Resource Management
    - Manage AP

Data Serialization

Intro

- Process of converting data into a standard format

JSON (JavaScript Object Notation)

- Whitespace insignificant
- 4 primitive data types
    - String
    - Number
    - Boolean: true/false, no quotes ("")
    - Null: null, no quotes
- 2 structured data types
    - Object/Dictionary: unordered list of key-value pair
    - Array: values don't have to be same types

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.1.1     YES manual up                    up
GigabitEthernet0/1     unassigned      YES unset  administratively down down
```

```
{
    "ip_interfaces": [
            {
            "Interface":"GigabitEthernet0/0",
            "IP-Address": "192.168.1.1",
            "OK?": "YES",
            "Method": "manual",
            "Status": "up",
            "Protocol": "up"
            },
            {
            "Interface":"GigabitEthernet0/1",
            "IP-Address": "unassigned",
            "OK?": "YES",
            "Method": "unset",
            "Status": "administratively down",
            "Protocol": "down"
            }
    ]
}
```

XML (Extensible Markup Language)

- Whitespace insignificant
- <key>value</key>
    - Similar as HTML format

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     192.168.1.1     YES manual up                    up
GigabitEthernet0/1     unassigned      YES unset  administratively down down


R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
  <ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
    <SpecVersion>built-in</SpecVersion>
    <IPInterfaces>
      <entry>
        <Interface>GigabitEthernet0/0</Interface>
        <IP-Address>192.168.1.1</IP-Address>
        <OK>YES</OK>
        <Method>manual</Method>
        <Status>up</Status>
        <Protocol>up</Protocol>
      </entry>
      <entry>
        <Interface>GigabitEthernet0/1</Interface>
        <OK>YES</OK>
        <Method>unset</Method>
        <Status>administratively down</Status>
        <Protocol>down</Protocol>
      </entry>
    </IPInterfaces>
  </ShowIpInterfaceBrief>
```

YAML (YAML Ain't Markup Language)

- Whitespace significant
- File starts with "---"
- "-" used to indicate a list
- Format -> key:value

```
---
ip_interfaces:
- Interface: GigabitEthernet0/0
  IP-Address: 192.168.1.1
  OK?: 'YES'
  Method: manual
  Status: up
  Protocol: up
- Interface: GigabitEthernet0/1
  IP-Address: unassigned
  OK?: 'YES'
  Method: unset
  Status: administratively down
  Protocol: down
```

REST APIs

API (Application Programming Interface)

- Software interface that allows 2 applications to communicate
- In SDN, API used to communicate btw
    - Apps and SDN controller (via NBI)
        - Usually REST API
    - SDN controller and network devices (via SBI)

CRUD

- Create, Read, Update, Delete
- Operations performed by REST APIs
- REST APIs usually use HTTP

HTTP verbs

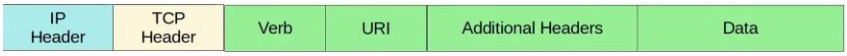
Purpose	CRUD Operation	HTTP Verb
Create new variable	**Create**	**POST**
Retrieve value of variable	**Read**	**GET**
Change the value of variable	**Update**	**PUT, PATCH**
Delete variable	**Delete**	**DELETE**

HTTP Request

- When HTTP client sends a request, the header includes
    - HTTP verb
    - URI (Uniform Resource Identifier) - indicates resource it wants to get

- URI

- Header



HTTP Response

- 1xx : informational
  - 102 Processing
- 2xx : Successful
  - 200 OK
  - 201 Created
- 3xx : Redirection
  - 301 Moved Permanantly
- 4xx : Client Error
  - 403 Unauthorized
  - 404 Not Found
- 5xx : Server Error
  - 500 Internal server error

REST (Representational State Transfer)

- Describe a set of rules on how API should work
- 6 constraints
  - Uniform interface
  - Client-server
  - Stateless
    - Each event is independent
    - If authentication required, need to authenticate in every single API request
  - Cacheable or non-cacheable
    - Not all resource, but those cacheable MUST be declared
  - Layered system
  - Code-on-demand (optional)

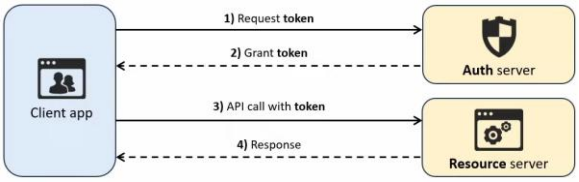
REST API Authentication

Intro

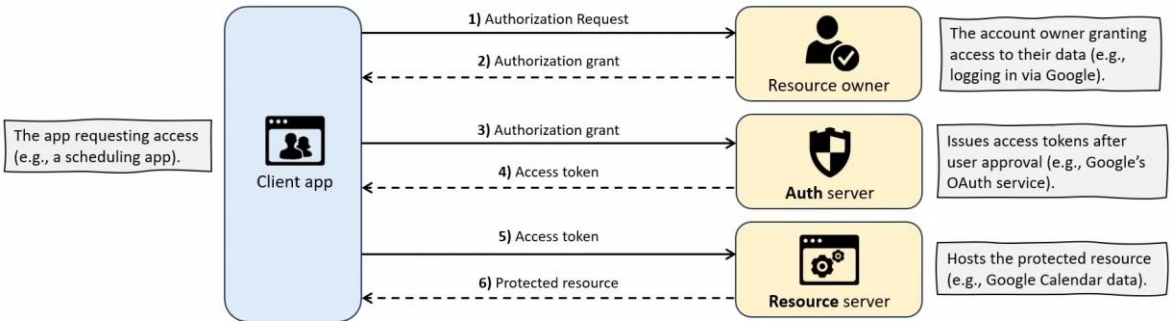
- Prevent attackers from accessing the data

Types of REST API Authentication

- **Basic Authentication**
  - Username/Password in Base64, NOT encrypted
  - HTTPS (TLS) for security
- **Bearer Authentication**
  - Use bearer token instead of username/password
  - Token expire after a short time



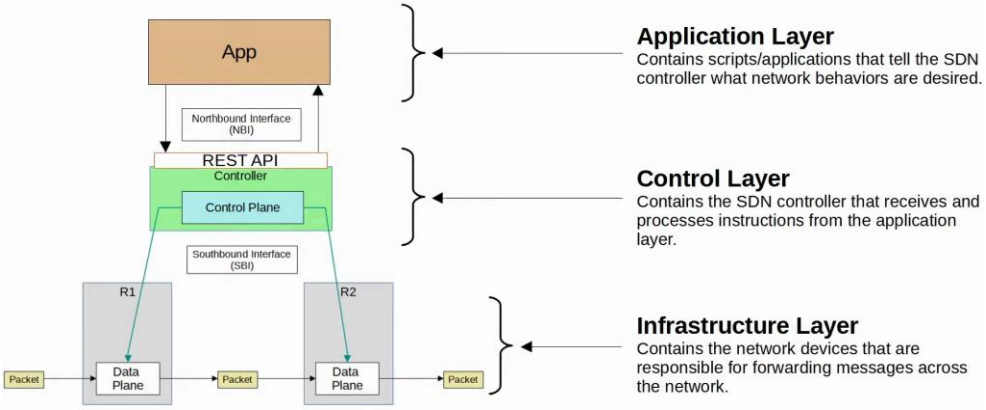
- **API key authentication**
  - Use static key issued by API provider
  - Good for tracking API usage
  - Easier to implement, but less secure
- **OAuth 2.0**
  - Provides **access delegation** - allow 3rd party apps limited access
  - Use **access token** that expire and can be refreshed (with a refresh token) w/o user reauthentication



SDN

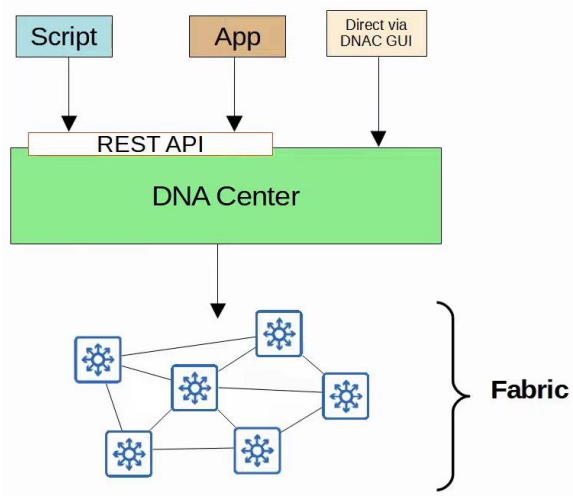
SDN (Software Defined Networking)

- Centralizes control plane into an app 'controller'
- Controller interact with network device using API



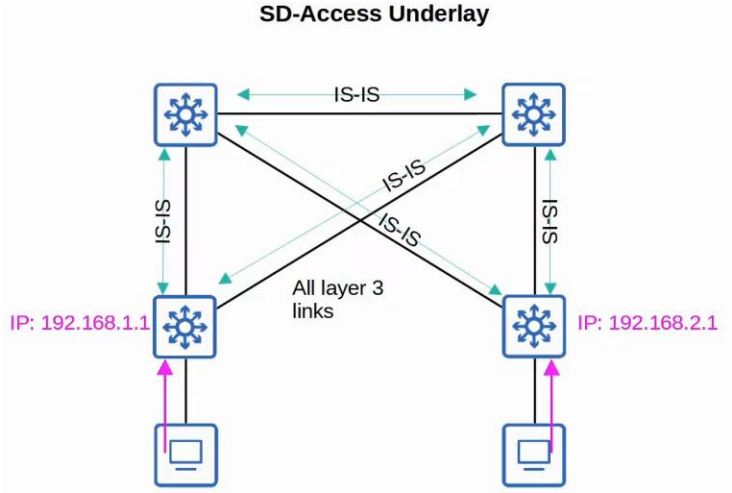
Cisco SD-Access

- Cisco's SDN solution for campus LAN
- Cisco DNA Center is the controller



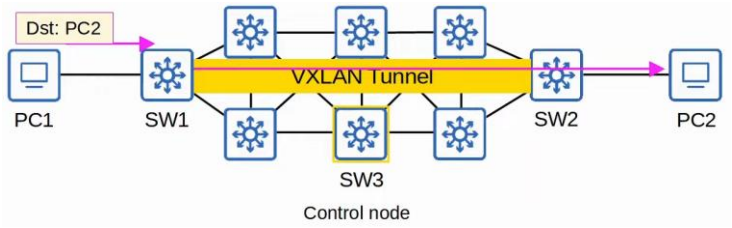
Underlay

- Underlying physical devices and connections
- Support VXLAN tunnels
- 3 roles for switches
    - Edge node: to end hosts
    - Border node: to devices outside of SD-Access domain
    - Control node: Use LISP (Locator ID Separation Protocol) to perform control plane functions
- New deployment
    - All switches Layer 3, use IS-IS
    - All links btw switches are routed ports, no STP
    - Edge nodes acts as default gateway of end hosts



Overlay

- LISP provides control plane of SD-Access
    - A list of EID and RLOC kept
    - EID (Endpoint Identifier): end points connected to edge nodes
    - RLOC (Routing Locators): corresponding end switch to end host
- Cisco TrustSec (CTS) provides policy control
- VXLAN provide data plane of SD-Access



Fabric

- Combination of the overlay and underlay
- The physical and virtual network as a whole

DNA Center vs Traditional Network Management

- Traditional
    - Device configured manually 1-by-1
    - Configs/policy per device
    - Network deployment long time
    - More error
- DNA Center
    - Centrally managed/monitored
    - Administrator give intent, DNA Center do the config on devices
    - Config/policy and software centrally managed
    - Network deployments faster

Configuration Management Tools

Intro

- Config drift
  - Config over time deviate from standard/correct config
- Config provisioning
  - How config are changed on devices
- Config management tools
  - Facilitate the centralized control of network devices

Ansible

- Written in Python
- Agentless: no need specific software on device
- Push model: push config to device
- Text files
  - Playbooks
    - Blueprint of automation tasks, YAML
  - Inventory
    - List devices and their characteristics, YAML, etc
  - Templates
    - Device's config file, Jinja2
  - Variables
    - Variables and their values, YAML

Puppet

- Written in Ruby
- Agent-based: need specific software on device
  - Not all Cisco device can
  - Can run agentless, using proxy agent
- Pull model: device pull from server
  - TCP 8140
- Files (proprietary language)
  - Manifest
    - Desired config state of network device
  - Templates
    - Similar to Ansible

Chef

- Written in Ruby
- Agent based
  - Not all Cisco device can
- TCP 10002
- Files use Domain Specific Language based on Ruby
  - Resources

        - 'Ingredients' , config objects
    - Recipes
        - Outline logic and actions
    - Cookbook
        - Set of related recipes
    - Run list
        - Ordered list of recipes that are run for device to achieve intended config

	Ansible	Puppet	Chef
Key Files defining actions	Playbook	Manifest	Recipe, Run-list
Communication Protocol	SSH	HTTPS (via REST API)	HTTPS (via REST API)
Key Port	22 (SSH port)	8140	10002
Agent-based/ Agentless	Agentless	Agent-based (or Agentless)	Agent-based
Push/Pull	Push	Pull	Pull

Terraform

Infrastructure as Code

- Managing/Provisioning infra using code instead of manual config
- IaC automate infra deployment/management

Provisioning vs Managing

- Provision
    - Build infra from scratch
- Managing
    - Manage already existing infra

Mutable vs Non-mutable

- Mutable
    - Resources updated
    - Config management tools

- Non-mutable
    - Create new version
    - Provisioning tools

Procedural vs Declarative

- Procedural
    - Explicit steps
    - Ansible, Chef
- Declarative
    - Define the end state
    - Terraform, Puppet

Terraform

- Provisioning tool
- Push model
- Agentless
- Workflow
    - Write - define desired state
    - Plan - verify changes
    - Apply - execute the plan/changes
- Written in Go
- Config file in HashiCorp Config Language (HCL)